TECHNISCHE
UNIVERSITÄT
DARMSTADT

Efficient Communication Concepts for Low-Earth-Orbit Pico-Satellite Formations

vom Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt

zur Erlangung des Grades
Doctor rerum naturalium
(Dr. rer. nat.)

Dissertation

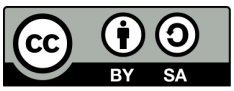
von Holger Ulrich Döbler

Erstgutachter: Prof. Dr. Björn Scheuermann

Zweitgutachter: Prof. Dr. Nils Aschenbruck

Darmstadt 2023

Holger Döbler:
Efficient Communication Concepts for Low-Earth-Orbit Pico-Satellite Formations
Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUprints: 2023
URN: urn:nbn:de:tuda-tuprints-243768
Tag der mündlichen Prüfung: 10.07.2023



Abstract

Advances in miniaturization during the last decades have enabled the construction of small spacecraft with total masses reaching down to 1 kg and below. At the same time standardization of components, interfaces, and platforms for the design of small satellites have further reduced their development costs. This trend has recently made formations of several small satellites in favor of one large unit an economically feasible alternative for university-scale organizations. While the satellite-formation approach provides new opportunities regarding Earth observation applications, it also poses new challenges to their communication system: the increased number of nodes at large distances and therefore propagation delays complicates the problem of medium access; the distributed collection of Earth observation data creates the necessity to gather the data over different links and possibly multiple hops. At the same time, the simultaneous operation of a number of identical satellites performing one task cooperatively can lead to redundancy in the data that needs to be communicated; exploiting this redundancy improves the overall efficiency of the communication system. In this work we discuss design approaches for communication protocols for Earth-observing satellite formations. In doing so we cover several layers from medium access control over network and transport up-to and including compression of payload data.

Cooperative position awareness beaconing is nowadays required for vessels at the high seas and serves as a convenient example for the in-orbit reception of data from large terrestrial sensor networks. We demonstrate how a purpose-made medium access control protocol can improve both, terrestrial performance for cooperative awareness and in-orbit overhearing for the purpose of global tracking.

Regarding higher network layers we discuss modern coding techniques like network coding and distributed source coding. These are less frequently used in terrestrial general-purpose communication networks like the Internet but can be employed to take advantage of the communication redundancy that is inherent to satellite formation operation. We show how use cases that differ in terms of kind of payload data and network topology each can benefit from their own, best-suited communication technique. We find that network coding is well-suited for over-the-air programming of satellite formations, that is, for ground-station-to-satellite broadcast transmissions. To adapt the concept of random linear network coding to typical topologies of satellite-and-ground-station networks, we

introduce a novel decoding algorithm that enables protocols to use simpler feedback mechanism.

Satellite-to-ground-station transmissions of multiple satellites' payload measurement data is often correlated across nodes. Here we demonstrate the applicability of distributed source coding techniques to increase the efficiency of communication resource utilization.

As a common bottom line that applies to all of these subtopics we conclude in the end that in Earth-observing satellite formations, there is a plethora of different types of information redundancy across the satellites. The exploitation thereof allows tailored communication protocols to significantly outperform their state-of-the-art terrestrial counterparts.

Zusammenfassung

In den letzten Jahrzehnten haben Fortschritte in der Miniaturisierung den Bau kleiner Satelliten mit Gesamtmassen von 1 kg und weniger ermöglicht. Währenddessen hat die Standardisierung von Bauteilen, Schnittstellen und Entwurfsplattformen die Entwicklungskosten für Kleinstsatelliten reduziert. Diese Tendenz hat dazu geführt, dass Satelliten-Missionen mit Formationen mehrerer kleiner Satelliten anstatt einem größeren Satelliten auch kostentechnisch für Organisationen wie Universitäten umsetzbar sind. Der Ansatz, Formationen von Kleinstsatelliten zu verwenden, bringt nicht nur neue Möglichkeiten für Erdbeobachtungsmissionen mit sich, sondern erzeugt auch neue Herausforderungen für die Kommunikationssysteme: Das Problem des Medienzugriffs wird durch eine Vielzahl von Knoten mit großen Distanzen und daher großen Signallaufzeiten verkompliziert; die auf mehrere Satelliten verteilt aufgenommenen Erdbeobachtungsdaten müssen über, möglicherweise unterschiedliche, Funkverbindungen und mehrere Hops zusammengetragen werden. Dabei kann das kooperative Sammeln von Messdaten durch mehrere Satelliten dazu führen, dass Redundanzen in den Rohdaten und damit in den Nutzdaten entstehen, die per Funk übertragen werden müssen; dies kann ausgenutzt werden, um die Effizienz dieser Übertragungen zu steigern. In dieser Arbeit beschäftigen wir uns mit Entwurfsansätzen für Kommunikationsprotokolle für erdbeobachtende Kleinstsatellitenformationen. Dabei decken wir verschiedene Netzwerkschichten ab, vom Medienzugriff, über Netzwerk und Transport, bis hin zu Komprimierung von Nutzdaten.

Der Einsatz von Funksystemen für den automatisierten Austausch von Navigationsdaten ist heutzutage für die Hochseeschifffahrt verbindlich vorgeschrieben. Sie sind ein gutes Beispiel für die Möglichkeit, Nutzdaten terrestrischer drahtloser Sensornetze im Erdorbit zu empfangen. Wir zeigen, wie sowohl die rein terrestrische Effektivität von automatisiertem Navigationsdatenfunk als auch der satellitengestützte Empfang der zugehörigen Funknachrichten zur globalen Seeverkehrsüberwachung durch ein geeignetes, maßgeschneidertes Medienzugriffsverfahren verbessert werden kann.

Was höhere Netzwerkschichten angeht, beschäftigt sich diese Arbeit mit modernen Kodierungstechniken wie verteilte Quellenkodierung und Netzwerkkodierung. Obwohl diese Verfahren derzeit in terrestrischen Allzweckkommunikationsnetzen wie dem Internet nur wenig Verwendung finden, können sie genutzt werden, um die angesprochenen Kommunikationsredundanzen auszunutzen. Für Anwendungsszenarien mit unterschiedlichen Nutzdatentypen und Netzwerktopologien zeigen wir, wie jedes einzelne davon von speziell

darauf abgestimmten Kommunikationsverfahren profitieren kann. So stellen wir fest, dass sich Netzwerkkodierung gut für over-the-air programming, also die Übertragung einzelner großer Dateien von einer Bodenstation an eine Vielzahl von Satelliten, eignet. Um die gängigen Netzwerkkodierungsverfahren auf dieses Problem anwenden zu können, haben wir einen neuen Dekodieralgorithmus entworfen, mit Hilfe dessen sich durch bessere Rückmeldungsinformationen einfachere Protokolle konstruieren lassen.

Messen verschiedene Satelliten die selbe Art von Daten, beispielsweise das Erdmagnetfeld, in ähnlichen Erdorbits, dann sind die Rohdaten oft zu einem gewissen Grad korreliert. Für diesen Anwendungsfall demonstrieren wir, wie sich verteilte Quellenkodierung einsetzen lässt, um die verfügbaren Kanalressourcen effizienter zu nutzen.

Als gemeinsames Fazit all dieser Unterthemen stellen wir fest, dass es im Bereich der Kommunikation in erdbeobachtenden Kleinstsatellitennetzen etliche unterschiedliche Arten von Informationsredundanz zwischen den Satelliten gibt. Nutzt man diese mittels je nach Anwendungsfall maßgeschneiderten Verfahren aus, lassen sich die jeweiligen etablieren Lösungen in Sachen Robustheit und/oder Effizient deutlich übertreffen.

Danksagung (Acknowledgement)

> **Fabienne:**
> *Whose motorcycle is this?*
> **Butch:**
> *It's a chopper, baby.*
> **Fabienne:**
> *Whose chopper is this?*
> **Butch:**
> *It's Zed's.*
> **Fabienne:**
> *Who's Zed?*
> **Butch:**
> *Zed's dead, baby. Zed's dead.*
>
> — **Pulp Fiction**
> (1994)

Auch wenn ich dieses Pamphlet selbst geschrieben habe, wäre der lange Prozess, in dem es entstanden ist, nicht ohne die Unterstützung zahlreicher anderer Menschen möglich gewesen.

Ich danke Björn Scheuermann für die Geduld, die er aufbrachte, für die diversen Anregungen und kritischen Fragen, mit denen er zur Präzisierung der erarbeiteten Inhalte beitrug, für die gemeinsame Arbeit an den publizierten Artikeln und natürlich für die Begleitung im gesamten nun im Abschluss befindlichen Prozess.

Ich danke Andreas Freimann für das ganze Wissen um Raumfahrttechnik, das ich in zahlreichen endlosen Telefonaten von ihm erhalten habe. Ohne diesen Austausch hätte ich nur schwer den notwendigen Überblick über den Gesamtkontext Satellitenkommunikation erhalten können.

Darüber hinaus danke ich allen Leuten, die mich in meiner Zeit am Lehrstuhl für technische Informatik an der HU begleitet haben. Den vielen anregenden und auch kritischen Diskussionen mit euch entstammen zahlreiche Ideen, die in dieses Werk eingeflossen

sind. Unter den Kollegen hervorheben möchte ich dabei Roman Naumann, Hagen Sparka und Steffen Tschirpke. Ohne Roman und Hagen wäre ich auf so manches Thema, das jetzt in der Arbeit behandelt wird, gar nicht oder erst viel später aufmerksam geworden. Steffen, die Zusammenarbeit mit dir hat mir immer viel Freude bereitet. Meine technischen Probleme hast du immer prompt und elegant gelöst. Ich kann mich zwar immer noch nicht für 3D-Drucker begeistern, aber sich bei solchen Themen durch dich von der Arbeit abhalten zu lassen war immer eine Freude. Ich bedauere immer noch sehr, wie früh du von uns gegangen bist.

Einen besonderen Dank spreche ich einigen der studentischen Hilfskräfte aus, die mir bei der Implementierung geholfen haben. Simon Dehlwes hat mit mir die Tiefen des AIS Standards ergründet und ihn in eine simulierbare C++ Implementierung gegossen, auch wenn er mit seiner Einschätzung zu Kreisen auf Kugeloberflächen falsch lag. Max Schlecht hat mir mit seinen (gar nicht so *schlechten*, wenn mir dieser Witz erlaubt sei) Implementierungen von sweep trackings, ROATSs und weiteren Unterproblemen viel Arbeit abgenommen. Johannes Lange hat mir ebenfalls viel unangenehme Implementierungsarbeit abgenommen, indem er sich mit ortsabhängigen Noise-Modellen, Extrapolation von Schiffs-Mobilitätsmodellen, und weiteren Problemen beschäftigt hat.

Über allem Austausch mit den sonstigen Angehörigen der HU steht selbstredend die mannigfaltige Unterstützung, die ich von Sven Hager erhalten habe. Von ihm konnte ich lernen, wie ansprechende Grafiken aussehen müssen und worauf man sich in den Texten konzentrieren sollte. Er war stets offen und tolerant, auch wenn andere Dinge als die eigene Kehle mit Faxe gewässert wurden. In unermüdlichem Arbeitseifer war er schon (oder noch) vor Ort, wenn man um sechs in der Früh das Büro betrat, und trotzdem half er auch bis Nachts um vier mit, die Ergebnisse des Tages zu rekontextualisieren und zu den Klängen menschenverachtender Untergrundmusik aus anderen Blickwinkeln zu bewerten. Und bei all dem hat er auch noch konstant versucht, mich zu einer zügigen Fertigstellung dieses Werks anzutreiben; sein Scheitern ist bedauerlich, doch ihn trifft keine Schuld.

Bei all der Unterstützung, die ich *in Adlershof* empfangen habe darf nicht außer Acht gelassen werden, dass ich durchweg vollen Rückhalt in meiner Familie und damit viel Unterstützung von zu Hause und meinen Eltern erhalten habe.

Mariella danke ich dafür, mir dieses Projekt von den Lebensumständen her überhaupt erst ermöglicht zu haben. Ohne den emotionalen Rückhalt, die Zuwendung und Liebe, und letztlich die Unterstützung zu Hause, hätte ich weder die Kraft noch den Willen gehabt, das Ganze doch noch zu Ende zu bringen. Dir gilt der größte Dank von allen.

Jana und Sven danke ich für ein hervorragendes Lektorat, durch das ich die Menge der stilistischen und orthographischen Fehler in dieser Arbeit erheblich reduzieren konnte.

Ich danke dem deutschen Brauereiwesen für die Bereitstellung etlicher Hektoliter besten Bieres, welches durch mich in diese Arbeit geflossen ist.

Mein abschließender Dank gilt Jan und Hein und Klaas und Pitt, denn die haben Bärte, die fahren mit.

Contents

Part 0

Prologue

Prooimion

0

> *And the moral of the story is that we had better regard —after all those centuries!— zero as a most natural number.*
>
> — **Edsger Wybe Dijkstra**

TL;DR *This chapter can be skipped.*

0.0 How to Read this Thesis

There is a saying that tools whose manual starts with "How to read this manual" tend to create more problems than any other tools [119]. Having used GNU Make to build this document, to run experiments for our evaluations, to compile our code, and for much more, we want to honor its original author, Richard M. Stallman, by including such a helpful section in our work. Hopefully, reading this document is as much of a pleasure as reading the manual of GNU Make [120].

This work comprises six *regular chapters* (Chapter 2 through Chapter 7). Each regular chapter contains the scientific content of one smallest publishable unit (SPU), i. e., the content of one paper. The regular chapters are grouped into two parts. Part I consists of chapters based on publications, whereas Part II contains unpublished findings. To rephrase it in a more content-centric dimension to cut: Part I is concerned with communication of Earth observation data towards satellite ground stations, while Part II deals with transmitting bulk files like software updates from satellite ground stations. Each regular chapter, roughly resembling one paper, follows more or less the de facto standard structure of network-conference publications: introduction, related work, proposed method, evaluation, and conclusion. Even though the ideal order of reading strongly depends on personal preference, we suggest the following reading schedule for each regular chapter:

0. Read the beginning of the intro until getting bored.
1. Skim over the conclusion.
2. Read the section about the proposed method thoroughly.

3. Look at the evaluation plots.

4. Read the rest of intro and evaluation and the related work only as needed.

There are two exceptions:

0. In order to grasp the main contribution of Chapter 7 it is probably sufficient to read the conclusion after looking at Fig. 7.11. After all, that chapter was only written to justify Chapter 6 in the context of this thesis' topic.

1. Chapter 6, in contrast, is most likely worth to be read entirely,

If the impression arises that Chapter 7's only purpose is to justify the presence of Chapter 6 which seems to be (in other respects) somehow unrelated to this thesis' topic: we can confirm that.

Regarding mathematical notation, we tried to stick to agreed-upon conventions wherever possible and to define it otherwise. In case of doubt, Section A.1 is a glossary of mathematical notation used.

Finally, we would like to note that Section 0.0 as well as the epigraphs at the chapters' beginnings were written tongue in cheek. We want to apologize for any potential irritation caused by its fatuity.

Ultimately we would like to kindly ask the reader for leniency regarding our poor style of writing throughout this document, including but not limited to, overlong sentences that, as we are aware of, are neither usual nor appreciated in scientific English writing, but nonetheless constitute our own preferred mode which was applied in writing every paper's first draft and somehow made it through to the final version of this thesis, especially in the chapters that are, despite contributing novel and significant findings, not based on published content and therefore never underwent the streamlining process of making the text ready for submission to peer-reviewed conferences.

0.1 Numbering Does Start at Zero, Indeed

According to the 1972 ACM A. M. Turing Award recipient Prof. Edsger Wybe Dijkstra, "numbering should start at zero"[121]. Taking this advice to heart, we tried to use zero-based indexing whenever applicable for mathematical objects within this work. While this may seem unorthodox, especially when employed to vector and matrix indices, (e. g., Chapter 6), we favor consistency over convenience. Consequently, we usually specify ranges of integers as half-open intervals $[0, n)_{\mathbb{Z}}$ (rather than closed intervals $[0, n-1]_{\mathbb{Z}}$), since the former notation possesses the handy property that the interval's cardinality

equals its bounds' difference, thereby avoiding (mental) off-by-one errors. However, ranges written with en dash (e.g., "2–5") are meant to include the upper bound.

0.2 Relation of this Document to Already Published Articles

Each of the Chapters 2–5 is largely based on a peer-reviewed conference full paper, containing not only the ideas, but also literally copied passages of text, explanatory figures, and plots. Since none of these papers were written by this work's author alone, some content of this thesis have flown from the pen of the other co-authors, both literal text passages and more abstract background work like design and implementation of evaluation scenarios and so forth.

B. Scheuermann, co-author of all four of these papers, contributed to each paper in terms of ideas for (protocol) mechanism modifications, critical pre-reviews helping to refine evaluation, guidance in stringent argumentation and presentation, as well as fine-tuning of textual representation.

Apart from that, Chapter 2 and Chapter 3 were entirely crafted by H. Döbler, including the initial ideas of the protocol mechanisms, implementation and evaluation thereof, as well as the overall textual and graphical presentation. However, we would like to note that the implementation of the automatic identification system (AIS) and self-organizing time division multiple access (SO-TDMA) protocol where done by S. Dehlwes (Studentische Hilfskraft) and that the original idea of solving the shortcomings of SO-TDMA in terms of orbital receivability by means of a new Medium Access Control (MAC) protocol might originate from H. Sparka.

[0], the foundation of Chapter 4, is a collaborative work of the Chair of "Informatics VII : Robotics and Telematics" at Julius-Maximilians-University Würzburg and the Computer Engineering Group at HU Berlin. To the best of our knowledge, the original idea of improving real-time capabilities of a traffic-monitoring satellite constellation by means of inter-satellite communication originates from K. Schilling and B. Scheuermann. A. Freimann and A. Kleinschrodt contributed the simulation and scenario framework, i. e., satellite mobility, aggregated vessel mobility and a sketch of a channel model. The inter-satellite forwarding and aggregation model together with implementation, the conceptual design and implementation of the evaluation including generation of plots were contributed by H. Döbler. The resulting text that was published and is now in large parts reproduced

in this thesis was a joint effort, where individual contributions are no longer separable reasonably.

Chapter 5 is based on [1] which in turn is based on H. Sparka's master thesis with the title "Von Picosatelliten im Formationsflug zur Erde: Effiziente Übertragung von Erdmagnetfelddaten". Consequently, a major share of this chapter originates from H. Sparka, including the initial idea as well as all the heavy lifting like distributed arithmetic coding (DAC) implementation, design and implementation of evaluation, and the foundation of textual and graphical presentation. H. Döbler informally supervised the original master thesis and a lot of the core decisions that led to the proposed protocol/algorithm and evaluation setup are a product of a cooperative process of H. Sparka and H. Döbler. The final publication as well as Chapter 5 still contain a large share of H. Sparka's original text and graphics, even though the final work load to bring the content to a peer-reviewed-publishable manuscript was then undertaken by H. Döbler and, as with all other publications, B. Scheuermann. In the end, we would like to designate at least 55% of this chapter as H. Sparka's contribution, even though the individual contributions are no longer precisely distinguishable.

Finally, all of these papers received review comments in response to their initial submission. As these comments were incorporated when preparing the camera ready versions of the manuscripts, the anonymous reviewers can also be seen as indirect contributors to the work presented here.

Chapter 6 and Chapter 7, on the other hand, are more or less entirely crafted by H. Döbler. However, we'd still like to point out that in the evaluation of EAGER we used an implementation of JOYCE and HNC that was kindly provided by R. Naumann. In addition, we'd like to thank J. Bauer for an introduction to related work on state-of-the-art OTAP protocols in wireless sensor networks that we wrote about in Section 7.1.

1 Introduction

> *Und jedem Anfang wohnt ein Zauber inne,*
> *Der uns beschützt und der uns hilft, zu leben.*
>
> — **Hermann Hesse**
> (German-born Swiss poet)

1.0 Motivational Overview

From the beginning of the space age with the launch of soviet Sputnik 1 in 1957 [122], virtually all artificial Earth satellites have been equipped with some sort of wireless radio transmitters and/or receivers. This is due to the fact that wireless radio communication (including optical links) is the only way to retrieve observation data from the satellites as well as to carry out maintenance and control tasks on the satellites. The only exceptions here are retrieval of reconnaissance data by ferrying cassettes of analog film back to the surface using Recovery Vehicles that were then actively caught midair by airplanes in the lower atmosphere, as is was for example done by the United States Air Force from 1960 to 1984 [7], as well as maintenance flights of manned spacecraft to orbital observatories like the Salyut, Mir, and ISS, or the Hubble Space Telescope. For obvious reasons these mechanical ferrying approaches require an enormous amount of ressources and infrastructure and are therefore quite expensive.

While the gross of low Earth orbit (LEO) satellites in and before the 1990s had masses greater than 100 kg and above [8], advances in miniaturization led to the development of smaller and lighter satellites, that are nowadays usually classified by their total mass ranging from *small satellites* of 100 kg to 500 kg over *micro*, *nano*, and *pico* satellites, down to *fempto satellites* with a wet mass smaller than 100 g [8]. The specification of the *CubeSat* platform in 1999 [9] has gained great attention by the academic community. CubeSats are satellites consisting of one or more cubic units with an edge length of 100 mm and a mass of up to 1.3 kg per cube, placing them at the lightweight end of the nano satellite range. While traditionally, each new satellite has been designed from scratch, the standardization brought with the CubeSat specification led to a plethora of now well-accepted interface definitions as well as a market of Space-Commercial-Off-The-Shelf (COTS) hardware that can be used to design and assemble nano satellites in a modular and therefore simple and

inexpensive way. When combining the low development and hardware costs to build a CubeSat with the nowadays commonly applied approach to launch the nano satellites as a secondary payload from a launch vehicle that is used to place a much heavier primary payload into orbit (in most cases a large satellite), the total mission costs of building and launching a basic one-unit (1U) CubeSat can be as little as a few thousand US doller depending on satellite components and orbit [10].

Another paradigm shift enabled by the low-cost availability of nano satellites is the trend to plan satellite missions as cooperative multi-nano-satellite formations at total missions costs still lower than comparable single-large-satellite missions launched some decades ago.

1.1 Selected Challenges and Opportunities in Multi-Nano-Satellite Low-Earth-Orbit Missions

The wireless communication in LEO missions of multiple nano satellites poses challenges and opportunities that, taken in isolation, apply to several well-known other communication networks as well, but whose joint treatment allows for solutions more efficient and better-suited than state-of-the-art approaches known from the related fields of wireless network research.

Just as in networks of large satellites, node-to-node distances of hundreds or thousands of kilometers lead to rather long per-link delays and high propagation path losses compared with most typical terrestrial wireless links. But compared to large satellites, an individual nano-satellite can be considered much less capable in terms of power budget, computational processing power, and system reliability. Since nano-satellite formations are networks of many low-performance, power-restricted, and unreliable nodes, they share many boundary conditions with Mobile Ad-hoc Networks (MANETs) and wireless sensor networks (WSNs). The latter are similar to Earth-observing multi-nano-satellite missions not only in terms of node characteristics, but also in the sense that the network's primary objective is the distributed cooperative acquisition, aggregation, and delivery of measurement data.

One important difference, however, between WSNs and MANETs on the one side and satellite formations on the other side is the dynamic nature of network topology: WSNs are generally considered to consist either of stationary nodes or of rather unpredictably moving nodes; [11] the latter is also a common assumption for MANETs. Satellite networks on the other hand feature a highly dynamic but long-term predictable and to some degree periodically changing topology which unifies the best of both worlds. While qualified

knowledge about future links enables reasonable routing decisions just as in the case of stationary nodes, the rapid movement of nodes allows exploiting data ferrying [12] approaches.

Even though all of these network types have in common that energy consumption is an important limiting factor for both computational power and communication capabilities, satellites are typically equipped with solar panels that ensure regular power generation at each node. Thus, power-saving strategies do not translate to the extension of nodes' overall lifetime, as seen in battery-powered WSN nodes [13]. Instead, a fixed portion of the power produced on average can be allocated for communication during mission planning, constituting the node's power budget that must not be permanently exceeded. However, there is also not necessarily an inherent benefit of permanently undercutting the anticipated budget.

Finally, another significant property of nano-satellite formations is the heterogeneity of link and node characteristics: satellite ground stations, usually connected to data centers, can generally be considered to be equipped with steerable high-gain antennas and electrical as well as processing power by orders of magnitude greater than what can be provided by nano satellites. These differences in processing power, together with different physical boundary conditions like background noise levels that differ between LEO and the surface and varying link distances create a wide range of different and often highly asymmetric links within the same satellite mission scenario.

1.2 Contributions and Thesis Structure

The communication demand in Earth observation satellite missions can roughly be divided into retrieval of the primary mission's payload data and data for maintenance and satellites' self-awareness. The primary mission payload data is simply the observation data that is being collected in orbit, i. e., the main purpose of the whole mission. It can range from any kind of imagery in case of remote sensing applications [14], via in-situ physical measurements, e. g., Earth magnetic field [15] (as in Chapter 5) to in-orbit overhearing of terrestrial communication networks like AIS or automatic dependent surveillance–broadcast (ADS–B) (Chapters 2–4). The latter case of overhearing is special in so far as the process of data collection itself can be seen as an initial hop of wireless communication in contrast to the former examples where data collection is rather a metrological challenge. In Part I we discuss approaches to improve the retrieval of Earth observation data using appropriate communication protocols, both in therms of effectiveness (i. e., retrieving more or better data) and in terms of efficiency (i. e., using less resources). The part's structure is ordered according to the direction of data flow (see Fig. 1.0): Chapter 2 is about improving

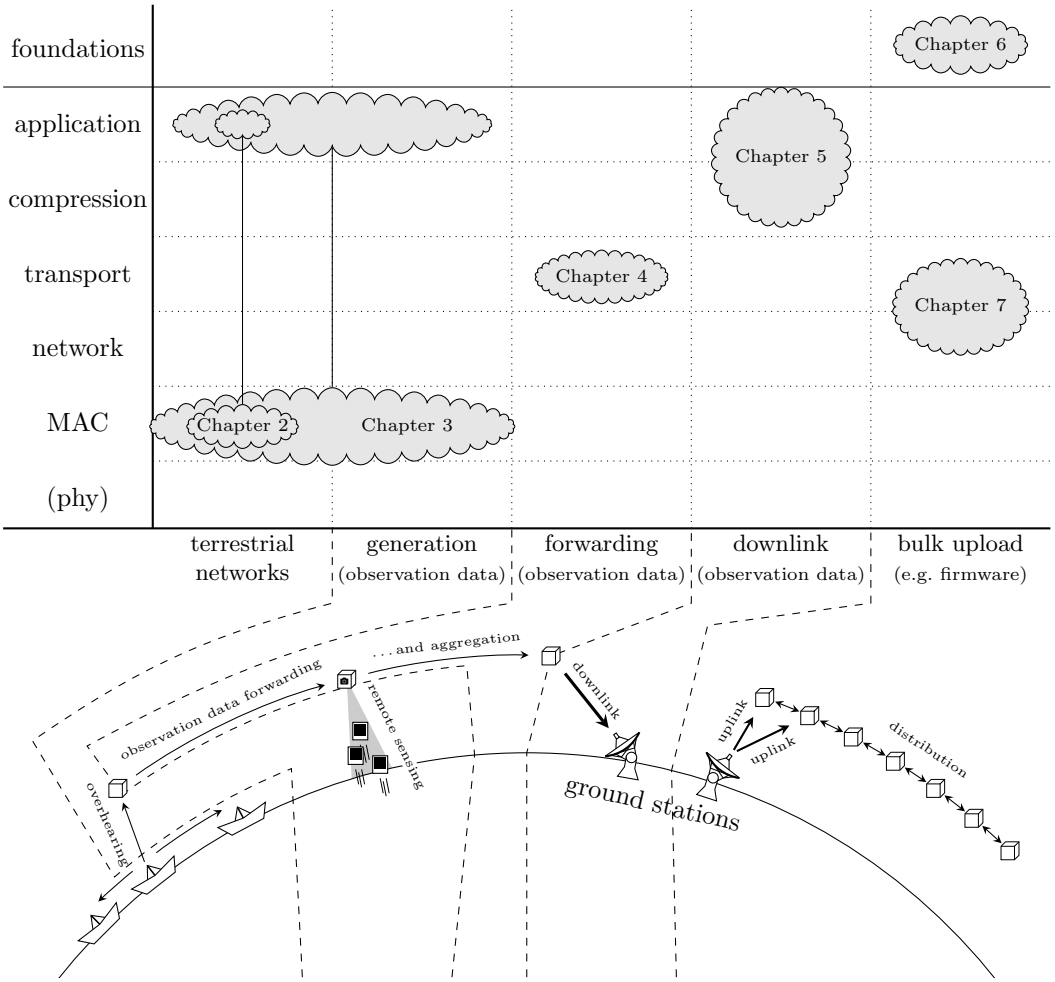


Fig. 1.0.: Thesis structure overview. Part I follows the path of primary mission payload data from the origin on the surface (Chapter 2) to the satellites (Chapter 3), via inter-satellite links (ISLs) in orbit (Chapter 4), and finally over a down-link to a ground station (Chapter 5). Part II covers other data flowing in the inverse direction, namely firmware updates and command-and-control data broadcasted from the ground station to the satellites. "Layers" on the chart's vertical axis do not follow any established layer model and are just descriptions to help classify contributions.

terrestrial cooperative awareness beaconing systems at the MAC layer by introducing the new location-assisted medium access for beaconing applications (LAMA) [3] approach that we find to out-perform AIS already in a purely terrestrial setting, without even considering satellites. This is followed by Chapter 3 that is concerned with tuning the medium access to receivability in LEO. In contrast to most of the related work on Satellite-AIS (S-AIS), we are not primarily concerned with how in-orbit reception of transmissions of the existing AIS can be improved, but instead propose a novel MAC protocol for terrestrial position awareness beaconing that is well-suited for passive in-orbit reception. In Chapter 4 we investigate the benefits of forwarding the received data in orbit via ISL, using a lightweight in-network-aggregation plus store-and-forward approach, in order to decrease traffic monitoring delay. Finally, the use of distributed source coding (DSC) to improve satellite-to-ground station downlinks is treated in Chapter 5. In a simulation-based case study we investigate the applicability of DSC in the context of simultaneous satellite-to-ground transmissions of physical measurement data from two satellites not communicating with each other.

Our contributions face varying boundary conditions in terms of available data to create evaluations scenarios, required fidelity of the simulated channel models, and even the applicability of the underlying techniques to the specific kind of payload data. This, among other reasons, led to different orbital configurations of the satellite formation, channel models, and mission goals (and thereby nature of payload data) being considered across the chapters of Part I.

Satellites are normally *not* operated in a fire-and-forget fashion. Depending on experience gained during the missions, software updates can become necessary. Satellites without a global navigation satellite system (GNSS) receiver might not be able to determine their orbital elements on their own, so this kind of operational metadata needs to be updated once in a while. Depending on the primary mission goal, orbital measurement campaigns need to be adapted or initialized during operation, e. g., when a customer suddenly orders imagery of a certain region. This task of reprogramming nodes is well-known as Over-the-Air Programming (OTAP) (or OAP) in the context of (terrestrial) WSNs.

Part II treats the broadcast transmission of bulk data, e. g., a firmware image, from a satellite ground station (GS) to a potentially large number of satellites. This part is structured differently, as it contains only one main contribution: the novel algorithmic approach EAGER to decode random linear network coding (RLNC) transmissions. EAGER, developed in Chapter 6, is applicable in a much broader context than just satellite OTAP. However, in Chapter 7 we demonstrate empirically that one unique feature of EAGER makes it particularly well suited precisely for RLNC-based OTAP for satellite formations, allowing

to achieve efficient and reliable broadcasts with protocols needing only simple feedback mechanism.

A conclusion is given in Chapter 8 which, together with the usual other back matter clobber, constitutes the epilogue.

Part I

Retrieval of Earth Observation Data

2 Location-Assisted Medium Access for Terrestrial Nodes

> *Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae, aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur.*
>
> — **Gaius Julius Caesar**
> from: Commentarii de Bello Gallico

This chapter is largely based on our own publication *"LAMA: Location-Assisted Medium Access for Position-Beaconing Applications"*[3] published as a full paper at MSWiM'19.

TL;DR *Two-hop-desynchronizing MAC for navigational-data broadcasts can efficiently be achieved by using the locally shared navigational data itself to arbitrate channel access.*

2.0 Introduction

Many of today's transportation systems achieve cooperative awareness by means of wireless beaconing. This includes ETSI ITS/WAVE for road traffic [123, 124], ADS–B for air traffic [125], and the AIS for maritime traffic [126]. All of these systems share the common feature that every vehicle periodically broadcasts small wireless beacons containing its geographical position, along with identification data and other navigational information. On the one hand these use cases share a set of challenging requirements for their wireless MAC protocol: the number of participating nodes is potentially large and distributed over a large area; nodes move, thereby constantly changing the network topology; all transmissions are link-local broadcasts, limiting the use of handshakes; successful, collision-free transmissions are safety-critical. On the other hand the use cases imply that nodes know not only their own position but also their neighbors' locations.

We propose to explicitly utilize the locally shared knowledge of node positions to resiliently avoid collisions while requiring only a small constant overhead per beacon. Our protocol *LAMA*, a **L**ocation-**A**ssisted **M**edium **A**ccess control protocol for beaconing applications, is especially suited to *large-scale* cooperative awareness applications. LAMA's basic idea is simple: in a slotted time, a distinct random *fire position* in the plane is assigned to each slot

by means of a hash of the slot number. For a specific slot, the node whose real position is closest to the fire position is allowed to use the slot for transmission while all other nodes must remain silent. Some extensions to this idea that allow for spatial reuse and improve channel access fairness are also discussed.

LAMA's advantage lies in the utilization of information that is locally shared between nodes anyway, as it is required by the primary use case itself. To the best of our knowledge, LAMA is the first MAC protocol for pure link-local-broadcast communication in dynamic topologies that achieves two-hop desynchronization, i. e., avoiding collisions also over longer distance and resolving the hidden terminal problem (HTP), without any neighbor state forwarding.

We have identified the maritime AIS as a well-fitting use case for the following three reasons: first, AIS is operated in the VHF band with high transmission power on the high seas, enabling typical communication ranges over 30 km. This leads to potentially high neighbor counts. Therefore, the impact of MAC protocol overhead is particularly strong in AIS, which may prohibit the use of more complex collision avoidance approaches. Third, due to the absence of large obstacles in the maritime environment, the channel's radio propagation properties are well predictable and homogeneous across different regions. LAMA can likewise constitute an interesting basis in other application areas, where cooperative awareness and tracking based on beacons is used. This includes, for instance, Intelligent Transportation Systems (ITS) or air traffic.

In our evaluation, we use large-scale maritime cooperative awareness as the use case and compare the implementation of our protocol to SO-TDMA, the MAC protocol used by AIS and therefore the state of the art in that field. In a comparative evaluation performed using the ns-3 discrete event simulator we show that LAMA outperforms SO-TDMA. The benefits are clear over a wide range of node densities, both in synthetic random topologies and for real marine vessel trajectory traces.

The remainder of this chapter is structured as follows. Section 2.1 gives an overview of the related work, especially discussing various MAC approaches. Our own approach is explained in detail in Section 2.2. In Section 2.3 we present a detailed performance evaluation of LAMA in comparison to SO-TDMA. Finally, a conclusion is given in Section 2.4.

2.1 Related Work

MAC protocols for wireless networks have been studied extensively. According to [16], MAC protocols for wireless networks can be either contention-based, like CSMA, or contention-free, like time division multiple access (TDMA). In some cases, a mixture of

both is used. The LAMA protocol proposed here is a distributed single-channel pure TDMA protocol. As LAMA is mainly a slot-allocation mechanism, it can be extended to multiple (sub-)channels and combined with orthogonal codes; such extensions are beyond the scope covered here, though.

Since beaconing for the purpose of position awareness is an inherently distributed use case, we skip the discussion of centralized protocols that rely on a base station, access point or head node that coordinates medium access of all nodes in range.

Distributed contention-based protocols either require significant additional resources like DBTMA [17] or suffer from a significant overhead due to control packets [18, 16, 19]. The same holds true for distributed hybrid MAC protocols such as HyMAC [20], which in addition assumes one or more base station (BS) nodes, acting as a data sink. Static contention-free MAC protocols that allocate fixed fractions of the resources available to each node are not applicable to networks of changing topology and unbound size.

Dynamic distributed contention-free MAC protocols typically map the MAC problem to a dynamic TDMA slot-allocation problem. Some of these protocols have *multi-channel* capabilities, i. e., they further divide each time slot in the frequency (frequency division multiple access, FDMA) or code (code division multiple access, CDMA) domain. In the Unifying Dynamic Distributed Multichannel TDMA Slot Assignment protocol (USAP) [21] nodes select unused TDMA slots and once in a while communicate their local slot allocation view to one-hop neighbors using special *control packets* in order to achieve two-hop desynchronization. In PTMAC [22] a similar approach is presented to resolve two-hop collisions in TDMA in the context of vehicular ad-hoc networks. This comes at the cost of MAC protocol overhead that scales with the number of neighbors.

Recent efforts to make the two-hop broadcasting of node states more efficient in the domain of vehicular ad-hoc network use Bloom filters [23]. They still require more overhead to broadcast Bloom filters than LAMA imposes. The Five Phase Reservation Protocol (FPRP) [24], as well as Evolutionary-TDMA [127], which uses FPRP for broadcast scheduling, use a five-way handshake in order to (re-)negotiate a broadcast slot assignment every time the topology changes. Thereby FPRP is applicable to dynamic, but slowly changing topologies. Despite the necessity for a significant amount of overhead, the FPRP mechanism relies on the nodes' capability to tell apart packet collisions from the absence of transmissions. This is a strong assumption, especially in a distributed setting. In SO-TDMA [126], on the other hand, no forwarding of neighbors' slot reservations is performed, thereby scaling well for large neighbor counts but suffering from the hidden terminal problem. LAMA requires neither handshakes nor other kind of control packets or state forwarding to achieve two-hop desynchronization, thereby generating less overhead.

Many existing approaches for cooperative awareness beaconing for road traffic in the absence of cellular network infrastructure are based on IEEE 802.11p or LTE-V2V sidelink mode 4 [25]. The former uses CSMA/CA and hence suffers severe performance degradation [26] in high node density settings. LTE-V2V (sidelink mode 4) on the other hand uses contention-free single carrier FDMA; nodes use Sidelink Control Information messages for reservation of resources similar to slot reservation in SO-TDMA. Instead of dedicated control messages, SO-TDMA, used by AIS, includes reservation information in every beacon header. In our performance evaluation, we compare LAMA against SO-TDMA as this protocol was designed specifically for small beacon sizes and data rates.

Our proposed MAC protocol LAMA is a distributed dynamic contention-free TDMA slot-allocation protocol that makes use of the nodes' locations to negotiate medium access without the need for explicit slot reservations. LAMA has a considerably small overhead of $O(\log_2(\#\text{ of neighbors}))$ bits per packet and uses no control, request, or confirmation packets at all. It does, however, require the nodes to know each other's locations approximately. In case of a transponder system whose primary goal is to broadcast location information, this is no overhead at all.

Medium access based on node locations has been used in geographic opportunistic routing, e. g., [27]. In contrast to LAMA, however, this work targets unicast routing; relative position information is used for (implicit) forwarder selection, whereas we use absolute positions for broadcast medium access coordination.

2.2 Location-Assisted Medium Access

2.2.0 Problem Statement

We propose a MAC protocol that is particularly suitable for large-scale cooperative awareness beaconing. Each of a large number of nodes distributed over a wide geographical area repeatedly broadcasts beacons containing its own navigational data. Each node is equipped with a radionavigation-satellite receiver providing it with the position information to be broadcast. Typically, this also provides clock synchronization, the other key ingredient for TDMA slot alignment. In addition, even though quite uncommon in distributed wireless settings, energy consumption of the communication system is not assumed to be an issue, because the vehicle's engines typically provide electric power on a much larger scale than what the communication requires.

In order to provide high quality positional awareness, nodes need to send beacons frequently, providing low-latency position information to their neighbors. At the same time interference on the wireless channel must be avoided.

2.2.1 The Basic LAMA Protocol

The basic idea of LAMA is to use the nodes' locations to mediate access to the wireless medium. This location information is particularly suitable for cooperative position awareness beaconing applications because up-to-date knowledge about neighbors' locations is inherent in this use case. LAMA is a single-channel TDMA protocol. Let us for the moment assume that at all times each node exactly knows all nodes' positions within a sufficiently large bounding box.[0] For each slot, let there be a so-called *fire position*: a 2D position, sampled uniformly from the bounding box, known by all nodes. For a certain slot, each node draws the same fire position; this can, e. g., be realized by using the same pseudorandom number generator (PRNG) for sampling the fire positions, or by hashing the time slot ID to a position. Fire positions for different slots are assumed to be statistically independent; using a good PRNG or a good hash function serves the purpose.

Using this mechanism, a simple medium access scheme—and the starting point for LAMA—is the following: *A node uses a slot for transmission iff it is closer to the fire position than all other nodes.* If nodes have perfect knowledge about all nodes' positions, this completely prevents collisions, because in every slot exactly one node sends. Perfect knowledge about all other nodes will of course not be given in practice. However, we will see that the performance degradation caused by typical position inaccuracies is very limited.

A remaining problem with this naïve first protocol is that it prohibits any spatial reuse, which results in poor channel utilization for networks covering larger areas. Let the length L be a parameter that describes the desired distance scale of spatial reuse. Instead of a single *fire position*, we assign a set of many fire positions to each slot that are mutually L or further apart. To maximize the number of fire positions in each slot, a hexagonal lattice of edge-length L can be used.

A node then selects the lattice point closest to its own position as the slot's *nearest fire position* (**np**) and uses the slot for transmission if no other node is closer to **np**. Apart from enabling spatial reuse, this gets rid of the necessity for a bounding box: the fire positions can now simply be drawn uniformly at random (UAR) from the lattice's unit cell in order to achieve a spatially homogeneous fire-position probability density across the entire plane. This mechanism can, however, lead to simultaneous transmissions of

[0] We will soon drop this assumption.

arbitrarily close nodes: two nodes aware of each other can correctly get different **np**s in the same slot, thereby each being closer to its **np** than the other one and hence both transmit. To prevent this, we define a *base send region* around each node's position and let a node transmit only if the **np** lies within this region (in addition to being the nearest node).

The protocol described so far can suppress most interference while enabling spatial reuse, once a steady state is reached where each node knows at least its direct neighbors. To improve bootstrapping and situations where new nodes join the network, nodes skip a constant fraction of their assigned slots probabilistically. When a node is about to transmit in a certain slot, it does so with probability $P^{\text{a}} \in (0, 1]$ and listens for unknown nodes otherwise. New nodes know their own position and therefore do not need any specific behavior in order to join: they transmit just like any other node when they are closest to the fire position. This comes with a (small) probability of collisions, but as we will see is perfectly bearable. Throughout our evaluation we used a value of $P^{\text{a}} = 0.95$, which is sufficient to allow joining of new nodes while at the same time not affecting channel utilization too much. However, we would like to note that the improved version of the LAMA protocol, the CAMELAMA protocol that is introduced in Chapter 3, does not use probabilistic listening in own slots but instead relies on a more sophisticated bootstrapping mechanism described in Subsection 3.2.2.

Exact definition of the (basic) LAMA protocol A node A at position $\mathbf{x}_A$ performs the following steps to decide if it transmits a packet in slot i:

- It deterministically draws a *base* fire position $\mathbf{p}_i$ from the lattice's unit cell uniformly at pseudorandom. All nodes compute the same $\mathbf{p}_i$ for the same slot i.

- $\mathbf{p}_i$ defines a hexagonal lattice $\mathbf{P}_i$ of fire positions.

$$\mathbf{P}_i = \{\, \mathbf{p}_i + jL \cdot (1, 0) + kL \cdot (\tfrac{1}{2}, \tfrac{\sqrt{3}}{2}) \mid j, k \in \mathbb{Z} \}$$

- From that, it computes its *nearest fire position* (**np**):

$$\mathbf{np}_i(\mathbf{x}_A) := \operatorname{argmin}_{\mathbf{p} \in \mathbf{P}_i} \|\mathbf{p} - \mathbf{x}_A\|_2$$

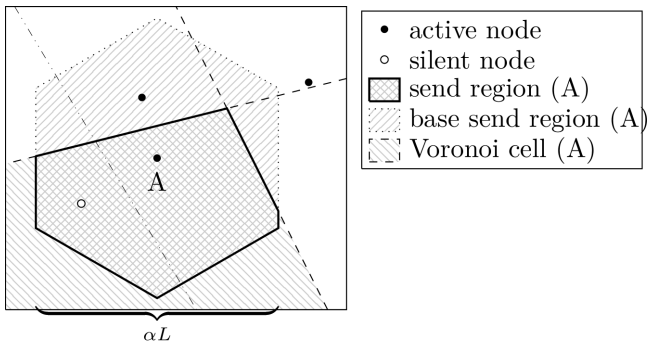


Fig. 2.0.: Schematic of the LAMA protocol mechanism. A node uses a slot for transmission only if a fire position lies within the node's send region.

- Node A sends in slot i if all conditions (2.0)–(2.2) hold:

$$\|\mathbf{np}_i(\mathbf{x}_A) - \mathbf{x}_A\|_2 < \|\mathbf{np}_i(\mathbf{x}_A) - \mathbf{x}_B\|_2 \,\forall\, B \in \{\text{other nodes}\} \tag{2.0}$$

$$\left|(\mathbf{np}_i(\mathbf{x}_A) - \mathbf{x}_A) \cdot (\cos\theta, \sin\theta)\right| < \frac{\alpha L}{2} \,\forall\, \theta \in \{0, \tfrac{\pi}{3}, \tfrac{2\pi}{3}\} \tag{2.1}$$

$$X_{[0,1)} < P^{\mathrm{a}} \tag{2.2}$$

where $\alpha \in (0, 1]$ and $P^{\mathrm{a}} \in (0, 1]$ are parameters of the protocol and $X_{[0,1)}$ is a uniform pseudorandom variable with range $[0, 1)$. For a visualization see Fig. 2.0. The first condition is fulfilled iff $\mathbf{np}_i(\mathbf{x}_A)$ is within the node's Voronoi cell. The second condition is fulfilled iff $\mathbf{np}_i(\mathbf{x}_A)$ is within the node's base send region, which we chose to be a regular hexagon with a minimal diameter of αL, centered at the node's position. For $\alpha = 1$ the base send region is the lattice's Wigner-Seitz cell shifted to the node's position and thus Condition (2.1) is always true. For $\alpha < 1$, Condition (2.1) assures that two nodes referring to distinct **np**s can only send in the same time slot if they are at least $(1 - \alpha)L$ apart. Condition (2.2) manifests probabilistic listening in own slots. We emphasize that in contrast to the globally chosen fire position $\mathbf{p}_i$, the $X_{[0,1)}$ of different nodes shall be statistically independent. In Fig. 2.1 we show a snapshot of an example simulation run of this simple variant of our protocol we call *basic LAMA*. This already yields good effective channel utilization and robustness as shown in Section 2.3.

2.2.2 De-Allocation of Slots

Basic LAMA is a zero-overhead collision-avoiding slot allocation mechanism. No information needs to be communicated beyond the node positions, which are the targeted applications' primary payload and are therefore exchanged anyway. Despite its efficiency, basic LAMA can lead to unfair, counterproductive allocation: first, it under-represents

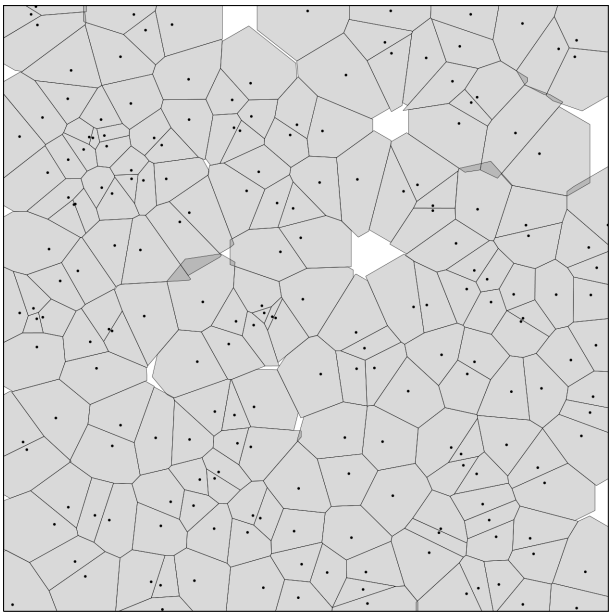


Fig. 2.1.: Node positions and send regions in basic LAMA. The send regions (light gray) nearly form a Voronoi diagram of nodes. Only few regions show uncovered gaps (white) and small overlaps (dark gray).

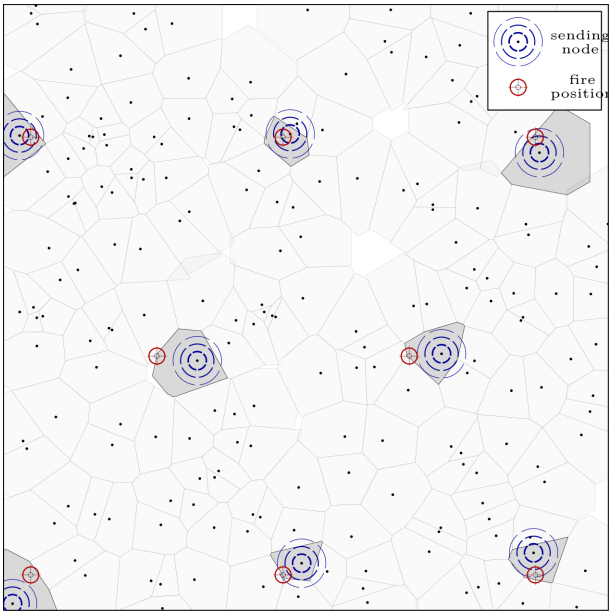


Fig. 2.2.: Sample lattice of fire positions and the resulting senders in the situation depicted in Fig. 2.1. The positions of sending nodes approximately replicate the hexagonal lattice structure.

nodes that are surrounded by very close neighbors. Second, each node's transmissions schedule resembles a discrete Bernoulli process. If the gap between successive beacons of a node is too short, the second beacon carries only little additional information. If that gap is too long, the position awareness accuracy suffers because of an increased position information latency.

We therefore introduce a simple mechanism that allows nodes to de-allocate future slots: each beacon's header contains a positive integer v that encodes a vow of silence that the source node will be silent in the following v slots. In this extended version of our protocol, that we call just LAMA, each node keeps track of its neighbors' silence state and only non-silent neighbors are considered when evaluating Condition (2.0). As a heuristic which value of v a node chooses to send, we use $v = \nu(k - \alpha^{-2})$, where k is the number of entries in the node's neighbor table and ν is a real-valued protocol parameter. For now, we can assume $\nu = 1$. In scenarios of homogeneous node density, a node with k neighbors in range should transmit every $k + 1$ slots on average. The additive term $-\alpha^{-2}$ is used to compensate for the bias that arises because a non-silent node needs to wait for the next transmission at least until a nearest fire position hits into its base send region. Since α^2 is the fraction of base send region area per lattice unit cell area, α^{-2} is the expectation value of the number of slots that a node needs to wait unit a **np** lies in its base send region. Even though $\nu = 1$ is the natural choice to achieve fairness, we will investigate the effects of this parameter in our evaluation. This mechanism induces a small amount of protocol overhead as $\nu \log_2 k^{\max}$ bits are needed to encode v in each beacon's header, where $k^{\max}$ is an upper bound of the number of neighbors a node is anticipated to observe at once.

Moving nodes So far we have not explicitly considered node movement, even though the main purpose of a beaconing application is to create up-to-date knowledge of *moving* nodes' positions. Every time a node sends a beacon, it includes its current position and saves this position. When evaluating Eq. (2.0), however, it uses the position it had sent in its most recent beacon instead of its current position, because its neighbors' state tables cannot contain more recent position information.

Non-planar geometry We assumed nodes to be in a flat plane, but vessels are located on the Earth's near-spherical surface. A spherical generalization of LAMA is straightforward, but requires replacing the hexagonal lattice with a maximal set of fire positions, pseudorandomly drawn for every slot, with a pairwise great-circle distance of at least L. While we constrain ourselves to a flat-plane approximation in this chapter, the improved MAC protocol CAMELAMA that is introduced in Chapter 3 treats node mobility and a fire position mechanism constrained to the surface of a sphere.

Forgetting nodes The protocol relies on each node's neighbor state table and its size. To prevent the table from growing indefinitely, nodes should delete neighbors from their table if no beacon was received for a certain amount of time. In our evaluation-based on real-world vessel movement traces, we experimented with neighbor expiration timeout values between one and 25 minutes, and found no significant impact on the protocol's performance.

2.3 Evaluation

We evaluated LAMA in the setting of AIS and compared it against SO-TDMA, the latter being the MAC protocol defined in [126] and designed particularly for that use case.

2.3.0 Simulation Setup

We implemented both LAMA and SO-TDMA for the AIS use case in ns-3 [128]. AIS uses VHF channels modulated in binary GMSK with $9600\,\text{bit/s}$. For TDMA slots of $\frac{2}{75}\text{s} \mathrel{\widehat{=}} 256\,\text{bit}$ are used. Each position report uses one slot and consists of roughly two bytes of slot reservation information and 19 bytes for the node's navigational state, including the sender ID. The remaining 11 bytes are used for header, trailer, and buffer time. SO-TDMA nodes explicitly reserve slots $1\,\text{min}$ to $7\,\text{min}$ ahead of time, avoiding slots reserved by other nodes. For further details of AIS and SO-TDMA we refer to the standard [126]. In our AIS implementation, we consider only position reports in continuous self-organizing mode of so-called "Class A" nodes,[1] a single channel, and a fixed reporting rate of 30 messages per minute, to avoid unnecessary protocol complexity that is hardly relevant for a MAC performance comparison. Our LAMA implementation uses exactly the same payload size and slot/frame structure; instead of the slot reservation data, the vow-of-silence value v is transmitted.

Packet loss was simulated using the signal-to-interference-plus-noise mechanism of *YansErrorRateModel* of ns-3, adapted to the characteristics of the maritime VHF channel: a SINR of $10\,\text{dB}$ results in a packet error rate (PER) of 20% [126]. For path loss and fading produced by Earth curvature and sea roughness, a two-log-distance model [28] was used with first path loss exponent $n_0 = 2.6$ according to [29] (with $3.75\,\text{m}$ antenna height, $1.5\,\text{m}$ sea surface height, $f_{\text{carrier}} = 162\,\text{MHz}$) and second path loss exponent $n_1 = 4.69$ [28] with transition distance $d_1 = 6.22\,\text{km}$ yielding a PER of 20% at $20\,\text{NM}$ (nautical miles) distance. Following [29], a normal random propagation loss with $\sigma = 0.65\,\text{dB}$ was added.

[1] Class A AIS transceivers are required for large vessels on international voyages.

To study the effect of different average node densities, we use the *random walk with reflection* mobility model, where nodes are confined to a 222 km square box. This is approximately six times the typical reception range in AIS. Inter-transition times and node speeds are drawn uniformly from $[0, 600\,\mathrm{s})$ and $[0, 30\,\mathrm{m/s})$ respectively. In the steady state node positions are distributed uniformly, and the node speed distribution equals the speed distribution at transitions [30].

We also use real AIS traces of vessels [129] that contain a set of position reports. Based on the trace of Jun 1st, 2017, 1:00–3:00 pm UTC, we applied a UTM transform to obtain Cartesian coordinates and cropped it to a $200\,\mathrm{km} \times 300\,\mathrm{km}$ rectangle to reduce the number of nodes to ≈ 550. Between resulting waypoints, constant velocities were simulated. To increase the real traces' node density, we took a cropped $400\,\mathrm{km} \times 400\,\mathrm{km}$ rectangle of traces with 904 nodes, and scaled time and space coordinates by a factor of $1/2$ to increase node density without reducing the nodes' speed.

Each measurement was performed with ten ns-3 simulation runs with independently seeded random number sequences. To shorten the time required for protocol bootstrapping, we did not start all nodes at the same time, but applied a staggered startup procedure where one node is added every 4 seconds. After all nodes joined the simulation, plus an additional initial equilibration period, data was measured for 30 simulated minutes in each run.

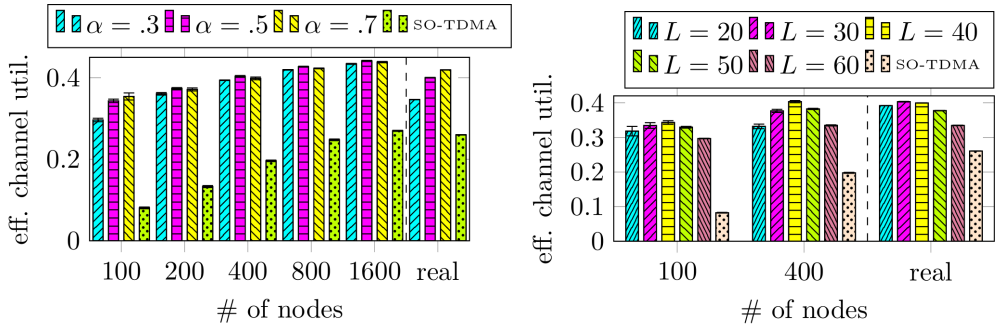
2.3.1 LAMA's Parameters

LAMA has three parameters that may systematically affect its performance: the lattice spacing L, the relative size of the base send region α, and the vow-of-silence scaling parameter ν. In the following, we explore their effects.

A simple robust performance metric is what we call the effective channel utilization (ECU): for a node, we defined it as the number of messages that a node received divided by the number of slots which the node itself did not use for transmission. Fig. 2.3a shows the channel utilization of LAMA for different values of α and different topologies. For $\alpha \in \{.3, .5, .7\}$, the channel utilization varies between 30% and 45% across all topologies considered. For the best value $\alpha = 0.7$, which we use for the rest of our evaluation, LAMA outperforms SO-TDMA by a factor of 1.6 to 4.4 in terms of ECU.

We also measured the effective channel utilization of LAMA for values of $L \in \{20, 30, 40, 50, 60\}$ nautical miles (NM). The results are given in Fig. 2.3b.

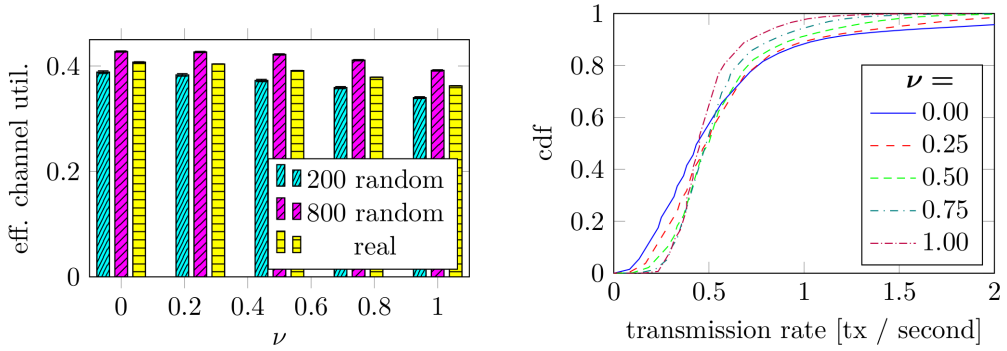
The effective channel utilization of LAMA is maximized for $L \approx 40\,\mathrm{NM}$ which is twice the maximum transmission range of the simulated model. However, the choice of L



(a) Random walk; different LAMA α-values with $L = 40$ NM vs. SO-TDMA.

(b) different LAMA lattice spacings (in NM) with $\alpha = .5$ vs SO-TDMA.

Fig. 2.3.: Effective channel utilization for different parameter sets. Error bars depict 95% confidence intervals (CIs).



(a) Effective channel utilization for different random and a real topologies. Error bars depict 95% CIs.

(b) Distribution of nodes' transmission rates in messages/second for 800 nodes random walk topology.

Fig. 2.4.: LAMA's performance depending on the choice of ν; $L = 40$ NM and $\alpha = 0.5$ are fixed.

has little influence on the performance. The robustness of LAMA with respect to L is a beneficial trait of the protocol. In maritime scenarios the transmission range may depend on environmental factors such as the weather and could vary over time or over different regions on Earth. Our results indicate that LAMA performs well in terms of effective channel utilization for a wide range of L.

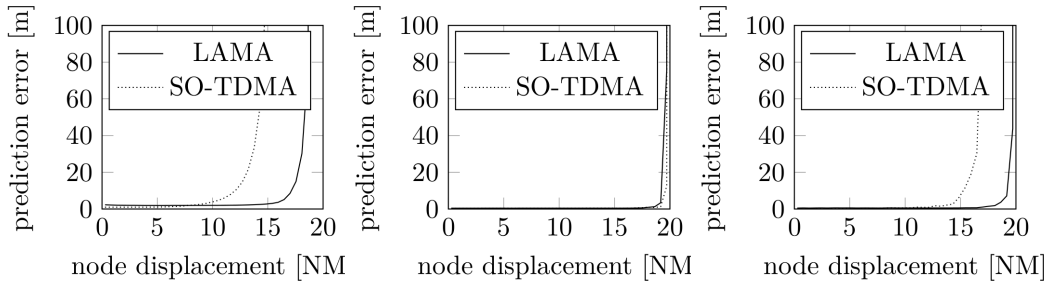
In Section 2.2.2 we introduced the vow-of-silence mechanism in order to improve the transmission rate fairness between nodes, as well as homogenization of inter-transmission times for each node. In order to quantify these effects, we varied ν in $[0, 1]$ for an 800-nodes random walk topology and measured the resulting effective channel utilization and the distribution of transmission rates. The effective channel utilization (see Fig. 2.4a) shows no significant dependency on ν, including $\nu = 0$ which corresponds to basic LAMA. The distribution of transmission rates however (see Fig. 2.4b) exposes that, as intended, higher values of ν reduce the width of the transmission rate distribution by effectively reducing the occurrence of both very high and very low transmission rates. This indicates that $\nu = 1$ is appropriate if transmission fairness is desired. This distribution was measured as follows: for a single simulation run, for every consecutive full minute (after the start-up phase) we measured every node's average transmission rate during that minute. Fig. 2.4b shows the cumulative distribution of these atomic measurements, combined from ten simulation runs.

2.3.2 Location-Prediction (In-)Accuracy

The goal of cooperative position awareness is to optimize the nodes' knowledge about each other's navigational state. Typically, low-latency mutual knowledge of navigational state is more important the closer nodes are. Therefore, as an application-oriented performance metric, we measured the nodes' position prediction errors as a function of the distance between two nodes: at a given time t, we recorded the sequence of tuples

$$\left(\overbrace{\|\mathbf{x}_A - \mathbf{x}_B\|_2}^{d(A,B)}, \overbrace{\|\mathbf{x}_A - \mathbf{y}_A(B)\|_2}^{\Delta_{\text{predict}}(A,B)} \mid \forall A \in \{\text{nodes}\}\ \forall B \in \{\text{nodes}\} \setminus \{A\}\right)$$

where $\mathbf{x}_A$ is the true position of node A at time t and $\mathbf{y}_A(B)$ is the position of A predicted by node B based on the last received message, containing A's position and velocity vector, using linear dead reckoning to extrapolate the position, course over ground, and speed over ground included in the beacon message received latest. We define $\Delta_{\text{predict}}(A, B) := \infty$ if B has never received a message from A. For a simulation run, we collected these samples for every ordered pair of nodes once every 7 s. The samples were then binned by $d(A, B)$ in consecutive bins of 1 km and for every bin we determined the 50% percentile of



(a) 1600 nodes random walk. **(b)** 541 nodes real. **(c)** 904 nodes scaled real.

Fig. 2.5.: Prediction error median as a function of node distance for LAMA and SO-TDMA in (a) a 1600 nodes random walk topology, (b) a real topology and (c) a scaled real topology. Error bars are omitted to avoid visual clutter.

$\Delta_{\text{predict}}(A, B)$. A comparison of LAMA @($L = 50\,\text{NM}$, $\alpha = 0.75$, $\nu = 1.0$) and SO-TDMA is given for a 1600-node random walk topology (Fig. 2.5a) as well as for the real and scaled real topology (Fig. 2.5b, 2.5c). In all scenarios and both methods, the median prediction error appears to be monotonic in the nodes' distance. It starts with a plateau ranging from 0 to 10 NM to 18 NM, depending on scenario and protocol, where the prediction error is $< 2\,\text{m}$. Each plateau is followed by a steep increase and error medians $< 100\,\text{m}$ are never observed for distances greater than 20 NM corresponding to the reception range of channel model. In case of the original real topology both protocols achieve a small prediction error for distances up to approximately 19 NM which we explain with a small overall node density that does not challenge the MAC protocol. In the other scenarios, however, we observe that the steep increase of the prediction error of SO-TDMA occurs at $\approx 4\,\text{NM}$ shorter distances compared with LAMA. The beginning of SO-TDMA's increase at 10 NM to 12 NM matches the fact that two nodes further apart than 20 NM are likely unable to desynchronize their transmissions. Thus, nodes in the middle between them suffer from HTP-type interference. A comparison for LAMA at different topologies (Fig. 2.6) shows that an increased node density mainly results in a prediction error curve shifted towards lower node distances.

2.3.3 Distance of Senders and Interferers

To visualize the effectiveness in avoiding collisions we measured the geographical distance of the sender to nearest simultaneously transmitting node for each transmission. Small distances to the next transmitting node are an indication for packet collisions while large distances indicate poor spatial reuse and therefore a waste of channel capacity. The cumulative distributions for LAMA @ $L = \{50, 60\}\,\text{NM}$ as well as SO-TDMA in a 400 nodes random topology are given in Fig. 2.7. The measurements where recorded over a time

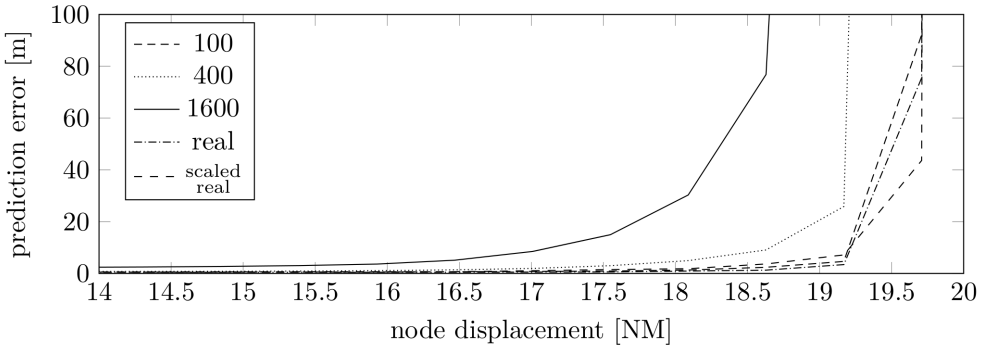


Fig. 2.6.: Prediction error median as a function of node distance for LAMA in different random walk topologies as well as real and scaled real topology.

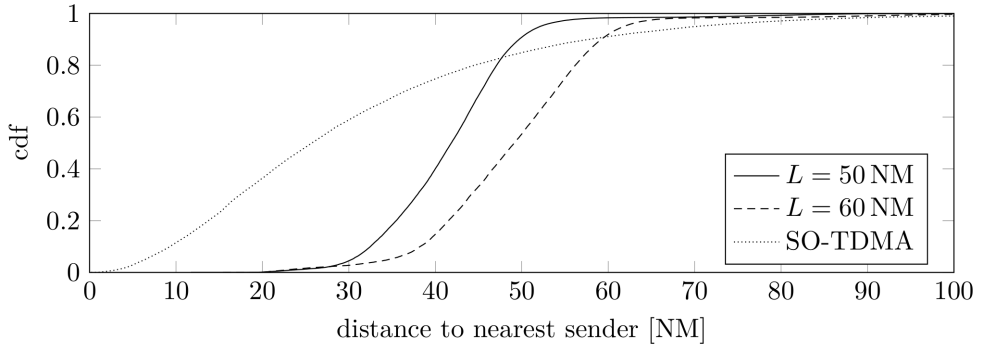


Fig. 2.7.: Cumulative distribution of the distance of each transmitting node to the nearest simultaneously transmitting node in a 400 nodes random topology.

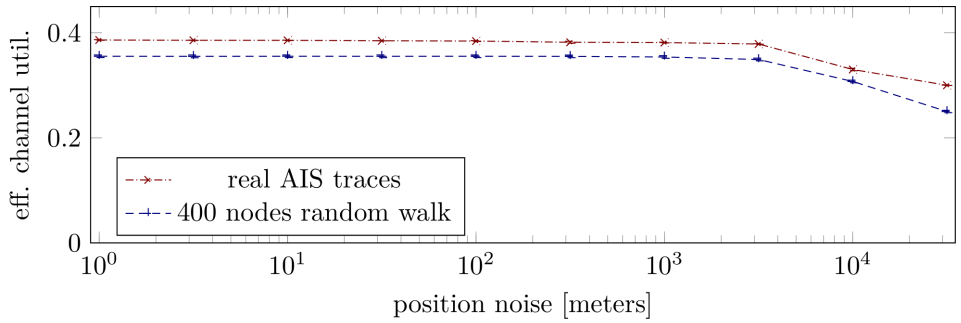


Fig. 2.8.: Average ECU achieved by LAMA with $L = 40\,\text{NM}$ depending on emulated node localization error. Error bars depict standard error measured in 10 runs.

of $\approx 14\,\text{min}$ and a total of $2.4 \cdot 10^5$ (LAMA, $L = 50\,\text{NM}$) / $1.9 \cdot 10^5$ (LAMA, $L = 60\,\text{NM}$) / $1.7 \cdot 10^5$ (SO-TDMA) transmissions have occurred in total. The corresponding distributions in case of LAMA are narrower than in case of SO-TDMA; 95% of the distance samples fall in 28 NM to 55 NM ($L = 50\,\text{NM}$) and 28 NM to 64 NM ($L = 60\,\text{NM}$) respectively. 55% of SO-TDMA's transmissions on the other hand where closer than 28 NM to the next sender producing both primary and HTP-type secondary interference, whereas either variant of LAMA scheduled only 2.5% of the transmissions within that distance range. This shows that even though LAMA was able to schedule more transmissions in total in the same time, it performed dramatically better in avoiding simultaneous transmissions of nearby nodes. As L represents the scale of spatial reuse, a distance slightly greater than twice the achievable or desired maximum transmission range seems appropriate.

2.3.4 Effect of Erroneous (Self-)Localization

So far we assumed perfect knowledge of each node's own position by means of a radionavigation device even though, e. g., for GPS positional errors of 1 m to 20 m are not uncommon [31]. To quantify the effect of this self-localization error we carried out a series of simulations with noisy position information: each time a LAMA device queries its node's position, it obtains the true position shifted by a pseudorandom noise vector. The noise vector's direction is drawn uniformly at random; the magnitude is the absolute value of a normally distributed variable with mean 0 and standard deviation A. The resulting average ECU for noise amplitudes $A \in [1\,\text{m}, 32\,\text{km}]$ for $L = 40\,\text{NM} \approx 74\,\text{km}$ is shown in Fig. 2.8. No significant effect is observable for realistic localization errors smaller than 1 km.

2.4 Conclusion

In this chapter we proposed the idea of LAMA, a radically new approach for a contention-free TDMA MAC protocol that efficiently achieves two-hop collision avoidance with only a small constant overhead per packet. LAMA is specifically well suited to the use case of transponder systems which require both short message sizes and high robustness with respect to topology changes. By combining a deterministic pseudorandom location-based slot assignment that is hard-coded into the protocol with a highly robust local arbitration mechanism, we completely circumvent the necessity for two-hop MAC state broadcasting. This distinguishes our approach from contention-based or contention-free MAC protocols known so far. The waiving of any state-forwarding makes our protocol inherently scalable with respect to the number of neighbor nodes as well as with respect to the total network size and diameter.

In our evaluation, we demonstrated LAMA's effectiveness to achieve high channel utilization and effective collision avoidance in synthetic random scenarios as well as for real-world vessel movement trajectories. Compared to SO-TDMA, which is the MAC protocol of the AIS, our protocol has shown to be superior in terms of position prediction accuracy as well as all other metrics that we considered. The remaining shortcomings of LAMA, namely its restriction to topologies that are geographically small enough to be considered as embedded in $\mathbb{R}^2$ and its lack of a robust bootstrapping mechanism are, among other topics, addressed by its successor protocol Cooperative Awareness and spaceborne Monitoring Enabled by Location-Assisted Medium Access (CAMELAMA) that is presented in Chapter 3.

3 CAMELAMA: Cooperative Awareness and spaceborne Monitoring Enabled by Location-Assisted Medium Access

> **Random dude:**
> *Christus war — soviel ich weiß — duldsam. Und wenn ihm einer widersprochen hat, dann hat er versucht, ihn zu überzeugen; hat nicht gesagt: „Halt deine Schnautze“*
> **Kinski:**
> *NEIN! Er hat nicht gesagt „Halt die Schnautze“, er hat eine Peitsche genommen, UND HAT IHM IN DIE FRESSE GEHAUEN!*
>
> — **Klaus Kinski**
> (Uraufführung „Jesus Christus Erlöser“, 1971)

This chapter is largely based on our own publication *“CAMELAMA: Cooperative Awareness and spaceborne Monitoring Enabled by Location-Assisted Medium Access”* [4], published as a full paper at WONS'22 as well as its extended version [5].

TL;DR *Using LAMA's ability to explicitly adjust the minimum distance between simultaneously transmitting nodes, orbital overhearability of navigational data beacon messages can be improved for a wide range of satellite swath widths without multi-hop propagation of state in the terrestrial network.*

3.0 Introduction

The primary purpose of navigational data beaconing systems such as AIS and ADS–B is to increase traffic safety by means of mutual cooperative position awareness of nearby

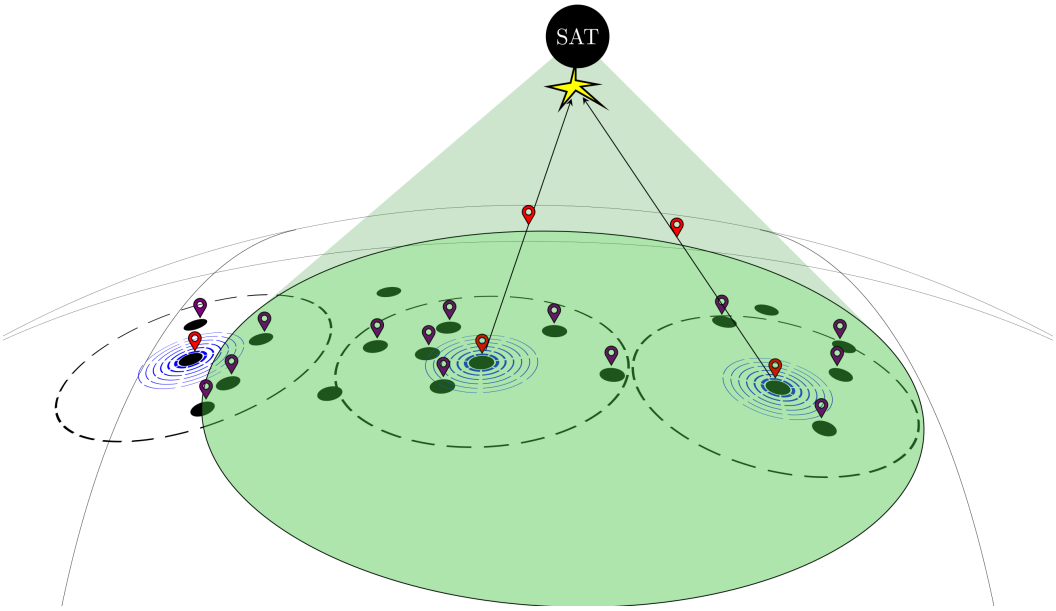


Fig. 3.0.: Schematic overview: The size of collision domains of terrestrial cooperative awareness beaconing networks (dashed lines) may be significantly smaller than those corresponding to LEO-based overhearing of these beacon messages (green cone/circle).

vessels. Nonetheless, it has been found that these messages can be overheard by LEO satellites to achieve global traffic monitoring [32, 33, 34]. Even though *single messages* can be received and decoded in LEO, collisions of multiple simultaneously transmitted messages are a severe problem in practice, because typical satellite antenna footprints span multiple terrestrial collision domains [35], as schematically depicted in Fig. 3.0.

In the past, this problem has been studied empirically and by modeling reception probabilities in S-AIS with respect to SO-TDMA, the MAC protocol of AIS that is mandatory for vessels at the high seas. Recent efforts to improve S-AIS performance were focused on advanced antenna and receiver design (see Section 3.1). We present an orthogonal approach and discuss, on the example of AIS, how a MAC protocol for cooperative awareness beaconing can be designed, if the satellite-reception use case is taken into account.

When assuming a high-gain receiver antenna in LEO with sufficiently small swath, terrestrial single-hop collision avoidance would imply good receivability in orbit. However, wider satellite antenna footprints are favorable for two reasons: first, it enables monitoring a greater surface area per time; second antennas with sufficient gain require greater antenna sizes not compatible with the size of nano satellites [36]. Two-hop collision avoidance at the surface on the other hand is not sufficient if a satellite's collision domain spans several terrestrial hops.

In this chapter we propose CAMELAMA, a MAC protocol based on the LAMA idea that is phy-layer-agnostic in the sense that it is merely a mechanism for the robust, decentral allocation of discrete units of channel resources (e. g., time slots) to nodes moving on the Earth's surface, while employing spatial re-use at appropriate length scales. CAMELAMA is designed to provide decent terrestrial cooperative awareness beaconing while at the same time enabling overhearing of beacon messages in LEO, i. e., desynchronizing transmissions in the corresponding larger collision domains.

One key ingredient for LEO overhearability of terrestrial beacon messages is that simultaneously sending nodes have a minimum distance to each other. The ability to set this minimum distance explicitly is already provided by the LAMA protocol, as can be seen for example in Fig. 2.7. However, in order to evaluate the performance with respect to spaceborne traffic monitoring, we changed and extended the LAMA protocol significantly in the following three aspects, which together result in the CAMELAMA protocol: first, while the geometrical foundation of the slot assignment mechanism requires nodes to "live" in a flat euclidean plane ($\mathbb{R}^2$) in case of LAMA, we changed that mechanism to nodes on the surface of the unit sphere (S_2) in CAMELAMA, as explained in Section 3.2.1. Second, naïvely tuning the minimum distance between simultaneously transmitting nodes to the orbital collision domain size would lead to a severe degradation of terrestrial cooperative awareness, thereby jeopardizing functioning of the MAC protocol itself. Therefore, CAMELAMA includes a straightforward interleaving mechanism introduced in Section 3.2.3 to afford transmissions intended for both in-orbit and terrestrial reception. Third, for protocol bootstrapping and to allow new nodes to join the network, LAMA uses a fairly simple mechanism of probabilistic listening, that is both lavish in terms of channel resources and yet so ineffective that it requires our simulations to start with a long period of staggered startup where nodes are carefully added one at a time. To make CAMELAMA more robust, we added a completely different and much more robust bootstrapping mechanism which is explained in Section 3.2.2.

This chapter also includes a quantitative empirical evaluation, where we modelled a large number of terrestrial nodes and a small number of LEO satellites using discrete event simulation (DES) and compared SO-TDMA with CAMELAMA in terms of orbital traffic monitoring performance. Our results that are presented in Section 3.3 indicate that especially for high node density, CAMELAMA significantly improves S-AIS performance.

Apart from that, this chapter contains Section 3.1 about related work regarding in-orbit reception of AIS beacon messages and a conclusion in Section 3.4.

3.1 Related Work

The problem of collision-free LEO reception of AIS messages has been studied extensively [37, 38, 36, 39, 40, 41, 42, 130]. In [42, 35] models of the AIS network were developed to predict the probability of receiving uncorrupted messages in LEO. [38] studies the impact of the orientation of satellites' monopole antennas on the receiving performance. In [37, 39] the performance of actually operating S-AIS missions is discussed. In several publications including [40, 36] designs of advanced S-AIS receivers are proposed to decode messages despite collisions using messages' differing Doppler shifts and propagation delays, soft decision decoding and Viterbi decoding, and successive interference cancellation (SIC). Others seek to lower effects of interference using digital beamforming [41]. These approaches take SO-TDMA as given and study and/or improve on the receiving side. CAMELAMA, in contrast, seeks to improve the medium access itself with respect to in-orbit receiving. As pointed out in Section 8, we believe that some of these approaches could be combined with CAMELAMA. While long range AIS [126, Annex 4] adds additional channel resources with the intent of satellite-based monitoring, it has no mechanism to desynchronize transmissions over multiple terrestrial transmission ranges.

Given that a LEO satellite's field of view (FoV) spans several SO-TDMA organized areas [34], the medium access pattern of vessels in distant organized areas looks approximately like random access. Several wireless random access MAC protocols that handle packet collisions using interference cancellation have been described [43]. Examples include slot-less access schemes by means of ZigZag decoding [44], but also the slotted flavor using the paradigm of coded random access [45] or generalizations thereof [46]. It is useful to counter the HTP [44], and is especially useful in situations where any sort of feedback is expensive, e. g., in device-to-device broadcast communication [47]. What these approaches have in common is the idea to introduce redundancy by spreading a transmission in time into multiple replicas of the same packet, either proactively or as the result of a retransmission mechanism. Decoding can then use this redundancy to iteratively extract information from collided transmissions, starting with a single non-collided transmission or part thereof. Like CAMELAMA, these protocols target the goal of feedback-less measures countering collisions of packets from hidden nodes. But the feedback-less subset of these SIC-based protocols lacks the ability to adapt transmission rates as it is needed for traffic beaconing scenarios where node densities change over time by orders of magnitude. Our approach guarantees a configurable upper limit of simultaneous transmissions per covered area.

With LT-MAC [48] and LBTM [49], other location-based TDMA protocols have been proposed. LT-MAC is designed for to-base-station unicasts whereas LBTM's goal is to reduce overhead of ACKs. As neither unicasts nor ACKs are needed for cooperative awareness, these approaches do not seem relevant in our context.

CAMELAMA re-uses some core ideas of the previously proposed protocol LAMA [3] and extends the protocol towards applicability for space-borne global traffic monitoring. LAMA is restricted to geographically confined networks whose topologies could be approximated in the Euclidean plane. CAMELAMA models a topology embedded on the unit sphere, enabling application of the protocol on a planetary scale. Where LAMA uses a simple hack to allow protocol bootstrapping and entry of nodes into the network that sacrifices channel resources for simplicity, CAMELAMA comes with a more robust mechanism. Finally, CAMELAMA can be adjusted to the aforementioned dual use as opposed to LAMA.

3.2 The CAMELAMA Protocol

3.2.0 Problem Statement

The protocol that is developed in this chapter is an extension of the LAMA MAC protocol from Chapter 2 not only in the sense that it is built around the same location-assisted slot allocation mechanism but also in the sense that one of CAMELAMA's two purposes is robust low-overhead MAC for high-seas vessel position beaconing applications. Therefore, the assumptions made in Section 2.2.0 apply here as well. In this chapter we add the secondary goal of in-LEO receivability of a significant share of these beacon messages in order to enable global traffic monitoring applications.

Our goal is to schedule fair and frequent transmissions of the vessels to achieve both: good cooperative position awareness between the vessels on the ground, and high packet reception rates for over-passing LEO satellites. However, we assume that nodes are not aware of the orbital elements or merely the number of overhearing satellites in orbit. To limit protocol complexity we also abstain from any active participation of the receiving satellite(s), i. e., we assume that the satellites are silent observers who do not influence the medium access protocol and are unknown to the sending nodes.

3.2.1 The (CAME)LAMA Slot Assignment Mechanism for Nodes on the Surface of a Sphere

The CAMELAMA slot assignment mechanism works similar to the mechanism used by LAMA: in every time slot i, a node A determines a set FP_i of slot fire positions and computes its nearest fire position $\mathbf{np}_i \in FP_i$ that is nearest to its own position $\mathbf{x}_A$. The set FP_i depends deterministically and pseudorandomly on the slot number i so that all

nodes agree on the same set FP_i. Node A sends in slot i if all of the conditions (3.0)–(3.2) hold:

$$\text{dist}(\mathbf{np}_i, \mathbf{x}_A) < \text{dist}(\mathbf{np}_i, \mathbf{x}_B) \,\forall\, B \in \{\text{other nodes}\} \tag{3.0}$$

$$\text{dist}(\mathbf{np}_i, \mathbf{x}_A) < r_{\text{max}} \tag{3.1}$$

$$\text{node } A \text{ is not in silent state} \tag{3.2}$$

where the radius r_{max} of the base send region is a parameter of the protocol. While the "base send region" in LAMA was a hexagon around each node's position, it is simply an open ball centered at the node's position in CAMELAMA. Setting its radius r_{max} to half the typical reception range of CAMELAMA messages implies that two nodes' base send regions overlap only if they are likely to be mutually aware of each other's positions. The mechanism by which nodes become silent for a certain time span is exactly the de-allocation mechanism used in LAMA and described in Section 2.2.2. LAMA's probabilistic listening mechanism is not used in CAMELAMA, as we explain in Section 3.2.2.

The main difference between LAMA and CAMELAMA lies in the domain used to model the nodes' positions and therefore in the geometrical nature of the fire position mechanism.

The LAMA protocol relies on a lattice (in $\mathbb{R}^2$) of fire positions to assign slot allocations that satisfy the desired constraints regarding minimum distance of simultaneously sending nodes as well as channel utilization. This hexagonal lattice, being a subset of $\mathbb{R}^2$, cannot be applied globally to scenarios where the nodes' positions are in first order confined to a sphere, as it is the case for vessels moving on the Earth's oceans (ignoring Earth's flattening for a moment). To find an analogous instrument for the pseudorandom assignment of fire positions on the sphere, let us look at first at the properties of the hexagonal lattice that make it a viable choice for the fire position assignment in *planar LAMA*.

0. For an explicitly given parameter L, every pair of lattice points is at least L apart.

1. Using a random displacement (i. e., translation drawn uniformly at random from the lattice's primitive cell), the areal probability density function of fire positions in a random time slot can be made constant.

2. Out of all subsets of $\mathbb{R}^2$ satisfying property (1), the hexagonal lattices with parameter L have the highest density of points (i. e., packing density).

3. It is computationally cheap to determine the lattice point that is closest to a given point in $\mathbb{R}^2$.

We note that only the first two properties are required for LAMA to function. The third property has an influence on the achievable channel utilization whereas the fourth property

has no impact on LAMA's performance regarding the metrics we evaluated so far, but seems favorable for the protocol's implementation on embedded hardware as well as for the computational demand of the simulations we use to evaluate LAMA.

We propose a mechanism that we call Randomly Oriented Approximative Tammes Sets (ROATS) to assign fire positions in CAMELAMA, i. e., for the LAMA protocol for nodes "living" on a spherical world. Let us from now on assume that in CAMELAMA all nodes' positions are elements of the unit sphere $S_2 = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\|_2 = 1\}$. Just as in *planar LAMA*, the fire positions shall be a subset of the same domain the nodes live in. We want to define the distance between two points in S_2 as the length of the geodesic between them or, equivalently, the angle between the first point, the origin, and the second point. We could as well have used the Euclidean distance in the embedding $\mathbb{R}^3$ $d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$ because the distance d_{S_2} uniquely depends on $d_{\mathbb{R}^3}$ strictly monotonically:

$$d_{S_2}(\mathbf{x}, \mathbf{y}) = 2 \arccos\left(1 - \frac{d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y})}{2}\right)$$

For a given $L \in (0, \pi)$, let $F \subset S_2$ be a set of points that are mutually at least L apart:

$$\forall \mathbf{x} \in FP_i,\ \mathbf{y} \in FP_i,\ \mathbf{x} \neq \mathbf{y} \colon d_{S_2}(\mathbf{x}, \mathbf{y}) \geq L \tag{3.3}$$

We call F the *base fire positions set*. In *planar LAMA*, the base fire positions set was a hexagonal lattice and the set of fire positions for a specific time slot is defined as the base fire positions set shifted (i. e., translated) by a pseudorandomly chosen displacement vector $\mathbf{v}$. Likewise in CAMELAMA, a rotation $R_i \in SO(3)$ is drawn uniformly at pseudorandom for each time slot i. The fire position set of slot i is then defined as $FP_i = \{R_i\mathbf{x} \mid \mathbf{x} \in F\}$. Hence, FP_i satisfies the same pairwise-distance constraint (3.3) as F does. In addition, if R_i was drawn uniformly at random from $SO(3)$, the fire positions' probability density function would be constant on S_2, implying that a node's retransmission rate does not depend on its absolute position but only on its position relative to other nodes.

So far we have not discussed how to choose F out of the many sets satisfying (3.3). In order to achieve a good channel utilization, the cardinality of F should be maximized. The problem how to place as many points as possible on a sphere while satisfying (3.3) is closely related to the Tammes problem [50]:

For a given number $k \in \mathbb{N}$, what is the maximum distance L such that there exists a set $F \subset S_2$ satisfying (3.3)*? And how can such a set be described?*

Unfortunately, exact solutions of the Tammes problem are currently only known for values $k \leq 14$ [51].

Most, if not all, state-of-the-art approaches to compute approximations to the Tammes problem for $k > 14$, as pointed out in [51], are numerical methods that place k *particles* on the sphere in randomly or systematically computed initial positions, assume a repulsive conservative force and then relax the system using simulated annealing, genetic algorithm, steepest descent methods, or other generically applicable optimization strategies. However, it turns out that numerical approximative approaches result in point sets with little regularity, which means that an implementation of LAMA would need to store the coordinates of all points of F explicitly. In addition, finding the closest point $f \in F$ for a given location $\mathbf{x} \in S_2$ either costs $O(|F|)$ time when using exhaustive search or requires an additional data structure to facilitate nearest-point search.

Let us use the parameterization (3.4) of S_2 where θ is called *latitude* and φ is called *longitude* as it is common for geographical problems.

$$\begin{aligned} x &= \cos\theta\,\cos\varphi & \theta &\in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}] \\ y &= \cos\theta\,\sin\varphi & \varphi &\in [0, 2\pi) \\ z &= \sin\theta \end{aligned} \tag{3.4}$$

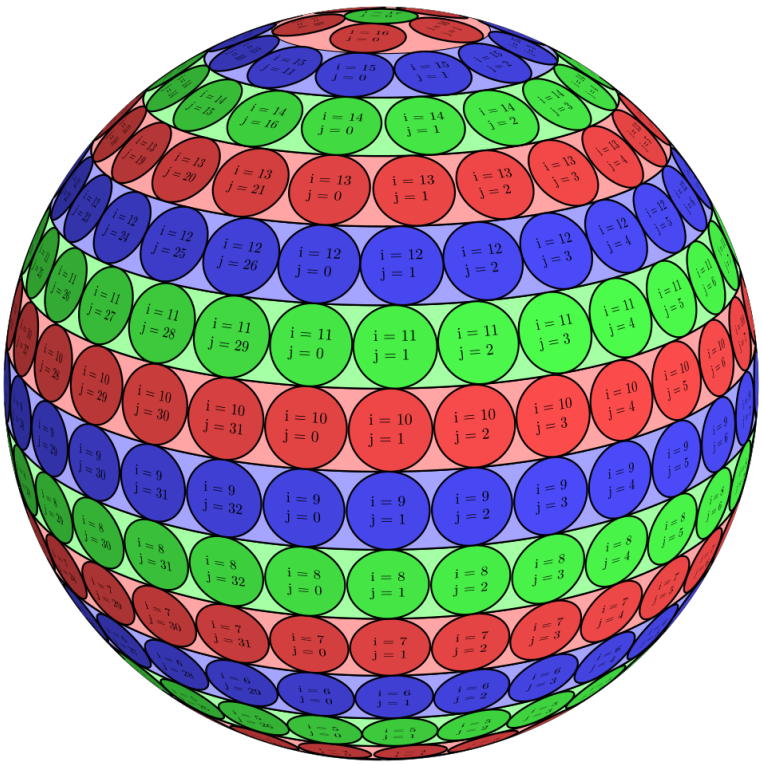
For the sake of simplicity of implementation we decided use base fire positions sets where the k elements are arranged in a small number (approximately $O(\sqrt{k})$) of layers; the points in each layer have the same latitude and are equidistant in longitude. Such sets are entirely described by

- N_{layers}, the number of layers
- $(\theta_0, \dots, \theta_{N_{\text{layers}}-1})$, the layers' latitudes
- $(n_0, \dots, n_{N_{\text{layers}}-1})$, the number of points in each layer
- $(\Delta\varphi_0, \dots, \Delta\varphi_{N_{\text{layers}}-1})$, a longitude offset for each layer

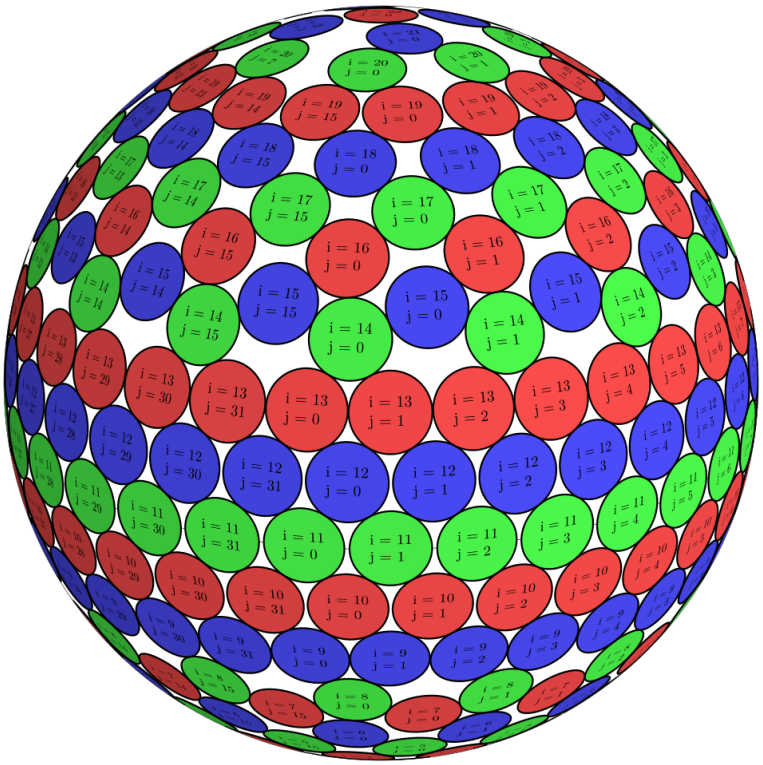
F is then constructed from these parameters by (3.5).

$$F := \bigcup_{i=0}^{N_{\text{layers}}-1} \{(\theta_i, \Delta\varphi_i + 2\pi\tfrac{j}{n_i}) | j \in [0, N_{\text{layers}})_{\mathbb{Z}}\} \tag{3.5}$$

A naïve choice for these parameters would be (3.6) where adjacent layers are exactly L apart in latitude and in each layer there are as many points as possible such that points inside a layer are mutually at least L apart (see Fig. 3.1a).



(a) Naïve packing.



(b) Greedy staggered by-latitude layering.

Fig. 3.1.: $L = \pi/17$ examples of naive (a) and greedy staggered by-latitude layering (b). The centers of the disks are located at $(\theta, \varphi) = (\theta_i, \Delta\varphi_i + 2\pi\frac{j}{n_i})$. The disks are shown with spherical radius $L/2$, so that the centers of two non-intersecting discs have a distance not smaller than L. Note that even though (a) might look denser, there are more disks placed on (b).

$$
\begin{aligned}
N_{\text{layers}} &= 1 + \lfloor \tfrac{\pi}{L} \rfloor \\
\theta_i &= (i + \tfrac{1 - N_{\text{layers}}}{2}) L \\
n_i &= \begin{cases} \left\lfloor \dfrac{\pi}{\arcsin(\frac{\sin \frac{L}{2}}{\cos \theta_i})} \right\rfloor & \text{if } \frac{\pi}{2} - |\theta_i| \geq \frac{L}{2} \\ 1 & \text{else} \end{cases} \\
\Delta\varphi_i &\equiv 0
\end{aligned}
\tag{3.6}
$$

Improved Layering

We use a slightly better (in terms of number of points placed for a given distance L) set of parameters that is generated by an algorithm we call *greedy staggered by-latitude layering*. Since we are not aware of an explicit formula of the resulting parameter values, we provide pseudo-code of the algorithm in Listing 3.0.

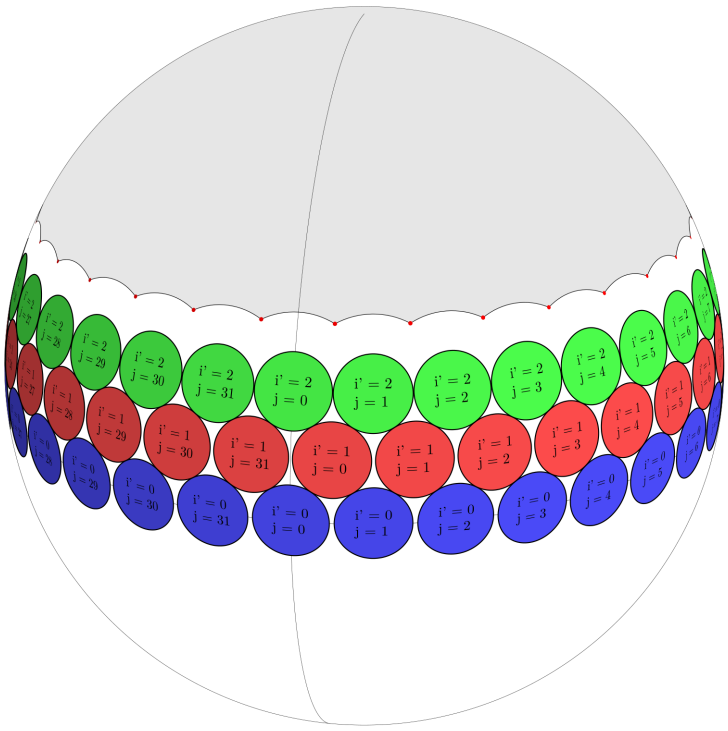
The algorithm can be summarized as follows: Initially the equator is filled with many points so that neighboring points are less than $2L$ apart. To populate the Northern Hemisphere, a new layer is always added as close to the equator as possible. A new layer has the same number of nodes as the last layer, if possible, or half that number otherwise. Using $k\,2^{\lfloor \log_2(2\pi/(kL)) \rfloor}$ equatorial points with $k \leq 7$ ensures that consecutively halving this number results in a sequence of integers until the pole is reached. If there is enough place of a single last point at the pole, it is also added. When the Northern Hemisphere is full, the southern hemisphere is populated analogously, starting with a layer that touches the equatorial layer and potentially the adjacent layer of the Northern Hemisphere. A visualization of the key concepts is given in Fig. 3.2. This algorithm results in a set of points F that are mutually at least L apart for given parameters L and k. Since we are free to choose k in order to maximize $|F|$ for given L, we compute $F(L, k)$ for $k \in \{1, 3, 5, 7\}$ and use the value of k that maximizes $|F|$. Larger values of k could have been used as well but rarely result in larger point numbers $|F|$.

Finally, we want to note that computation time required to compute F is by no means performance-critical, because it needs to be done only once while establishing protocol parameters for a specific protocol instantiation. The sequence of tuples $(\theta, n, \Delta\varphi)_0, \ldots, (\theta, n, \Delta\varphi)_{N_{\text{layers}}-1}$ is hardcoded into the protocol.

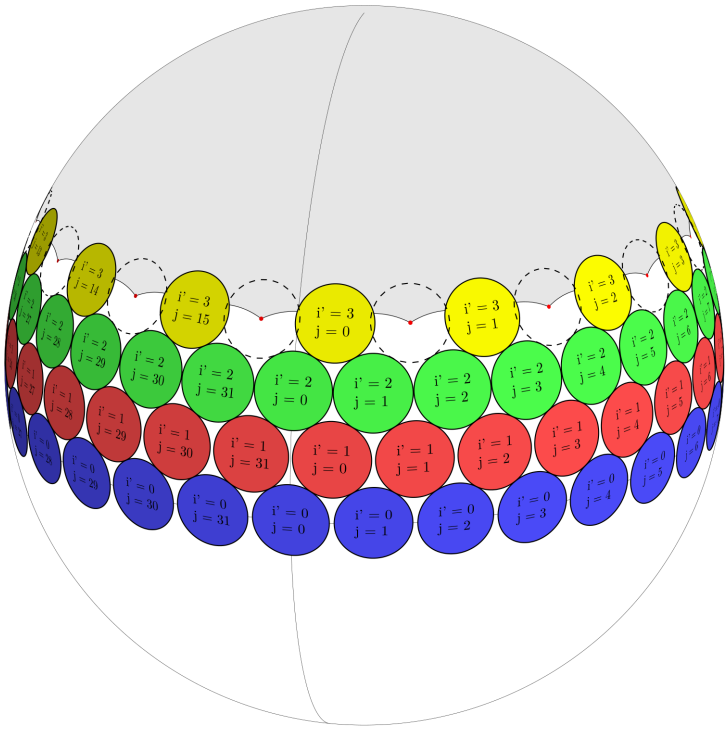
0 **function** LAYER_PARAMETERS(L)
1 **return** $\max_{\#\text{ points}}\{\,\text{LAYER\_PARAMETERS\_INNER}(L,k) \mid k \in \{1,3,5,7\}\,\}$
2 **function** LAYER_PARAMETERS_INNER(L,k)
3 $N_{\text{layers}} \leftarrow 1$
4 $\theta_0 \leftarrow 0$
5 $\Delta\varphi_0 \leftarrow 0$
6 $n_0 \leftarrow k\,2^{\lfloor \log_2(2\pi/(kL)) \rfloor}$
7 $n \leftarrow n_0$
} place first layer on equator
8 **while** $n \geq k$ **do**
9 $U \leftarrow \left([0,\frac{\pi}{2}] \times [0,2\pi)\right) \setminus \left(\bigcup_{i=0}^{N_{\text{layers}}-1} \bigcup_{j=0}^{n_i-1} B_L(\theta_i, \Delta\varphi_i + j\frac{2\pi}{n_i})\right)$
10 $C \leftarrow \min_\theta U$
11 $(\theta,\varphi) \leftarrow \min_\varphi C$
12 **while** $\text{dist}_{S_2}((\theta,0),(\theta,2\pi/n)) < L$ and $n \geq k$ **do**
13 $n \leftarrow \frac{n}{2}$
14 **if** $n \geq k$ **then**
15 $N_{\text{layers}} \leftarrow N_{\text{layers}} + 1$
16 $\theta_{N_{\text{layers}}-1} \leftarrow \theta$
17 $\Delta\varphi_{N_{\text{layers}}-1} \leftarrow \varphi$
18 $n_{N_{\text{layers}}-1} \leftarrow n$
19 **if** $\theta_{N_{\text{layers}}-1} + L \leq \frac{\pi}{2}$ **then**
20 $N_{\text{layers}} \leftarrow N_{\text{layers}} + 1$
21 $\theta_{N_{\text{layers}}-1} \leftarrow \frac{\pi}{2}$
22 $\Delta\varphi_{N_{\text{layers}}-1} \leftarrow 0$
23 $n_{N_{\text{layers}}-1} \leftarrow 1$
} add point at pole
24 Fill southern hemisphere analogously to lines 7 through 23.
25 Simultaneously sort (θ), (n), and $(\Delta\varphi)$ such that (θ) is monotonically increasing.
26 **return** $(N_{\text{layers}}, (\theta), (n), (\varphi))$

Where B_L is the open ball with respect to the spherical distance:
$B_L(\theta,\varphi) := \left\{p \in [-\pi/2,\pi/2] \times [0,2\pi) \mid \text{dist}_{S_2}(p,(\theta,\varphi)) < L\right\}$

Listing 3.0.: The algorithm by which the parameters of *greedy staggered by-latitude layering* are computed.



(a) The equator (blue, $i' = 0$, lines 3–6) and two more layers are shown. U (line 9), the subset of the Northern Hemisphere that is at least L apart from every point added so far, is drawn gray. Its southernmost points C are marked with red dots.



(b) A new layer (yellow, $i' = 3$) was added (lines 15–18). Since adding every red dot would have led to overlapping discs (dashed circles), the number n of points per layer needed to be halved (lines 12, 13).

Fig. 3.2.: A visualization of LAYER_PARAMETERS_INNER($L = \pi/17, k = 1$) in Listing 3.0 while layers are added to the Northern Hemisphere. Layer indices i' are changed by sorting (line 25) before the algorithm returns.

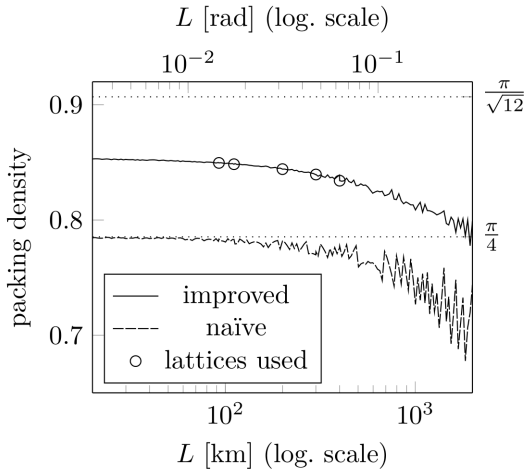


Fig. 3.3.: Packing efficiency of naïve and improved layered fire position sets. Naïve packing seems to approach $\frac{\pi}{4}$ for small L whereas improved layered packing achieves densities around 0.85 for the range of L we employ in our evaluation, which is approximately 7% smaller than the upper bound [53] $\frac{\pi}{\sqrt{12}}$.

Packing Density

We evaluate the achievable cardinality of the different point packing strategies in terms of their densities [50], i. e., the total area covered by spherical caps of radius $L/2$ at each point of F divided by the area of the unit sphere (3.7).

$$D_{L/2}(F) = \frac{|F|(1 - \cos\frac{L}{2})}{2} \tag{3.7}$$

In $\mathbb{R}^2$ the packing efficiency[2] of the square lattice is $\pi/4$ whereas the optimal packing efficiency of $\pi/\sqrt{12}$ is achieved by the hexagonal lattice [52]. A comparison of the packing efficiencies of both layered approaches that we consider is plotted in Fig. 3.3 for $L \in [20/6371, 2000/6371]$ (corresponding roughly to 20 km – 2000 km distance on our Earth).

Nearest point lookup

A CAMELAMA node needs to determine its nearest fire position in every slot, i. e., the element of PF_i that has the smallest distance to $\mathbf{x}$, the last own geographical position it transmitted in a beacon message.

[2] For infinite lattices a packing efficiency can be defined in terms of disk-coverage of their primitive cells.

In a first step, this problem is reduced to the problem of finding the point $f \in F$ that is nearest to $\mathbf{x}' := R_i^{-1}\mathbf{x}$:

$$f := \min_{\text{dist}(\cdot, R_i^{-1}\mathbf{x})} F$$
$$\Rightarrow \quad R_i f = \min_{\text{dist}(\cdot, \mathbf{x})} F P_i$$

because distance is invariant under rotations. In our implementation, the layer parameters $(\theta, n, \Delta\varphi)_0, \ldots, (\theta, n, \Delta\varphi)_{N_{\text{layers}}-1}$ are stored in an array, sorted by values of θ.

The point of layer $j \in [0, N_{\text{layers}})_{\mathbb{Z}}$ that is closest to $\mathbf{x}'$ is simply the longitudinally closest point given by (3.8).

$$\begin{aligned} \theta(f_j) &= \theta_j \\ \varphi(f_j) &= \Delta\varphi_j + \left\lfloor \frac{1}{2} + \frac{(\varphi(\mathbf{x}') - \Delta\varphi_j) \cdot n_j}{2\pi} \right\rfloor \cdot \frac{2\pi}{n_j} \end{aligned} \tag{3.8}$$

Even though one could search exhaustively through all layers to find f, we found empirically that the approach described in Listing 3.1, which searches only in a much smaller subset of layers, is significantly faster, despite yielding the same result.

```
0 function FIND_NFP(x')
1     j ← ⌊N_layers · (1/2 + θ(x')/π)⌋
2     while j ≥ 0 ∧ θ_j > θ(x') do
3         j ← j − 1
4     while j + 1 < N_layers ∧ θ_{j+1} ≤ θ(x') do
5         j ← j + 1
6     return f_k nearest to x' according to (3.8) for k ∈ [j − 2, j + 3]_ℤ
```

Listing 3.1.: An algorithm to compute the point $f \in F$ nearest to a given point $\mathbf{x}'$ without exhaustively iterating over all layers.

3.2.2 CAMELAMA's Bootstrapping Mechanism

One of the worst properties of the LAMA protocol is its hostility with respect to new nodes. In basic LAMA, nodes that join the network are only allowed to transmit within the slots that where previously assigned to their nearest neighbors. In order to let a node introduce itself to its neighbors, we used the mechanism of probabilistic listening in own slots. But this mechanism looks rather like a naïve hack in an otherwise elegant protocol:

- It reduces the channel utilization by a constant factor.

- It does not allow nodes to enter the network without intentionally creating packet collisions.
- It does not work at all if too many nodes enter the network simultaneously within a small geographical area.

Because of the last point, we needed to use a long-stretched staggered startup procedure in the simulations of our evaluation. In scenarios that we simulate to study overhearing by satellites we needed to go to sizes of several thousand nodes per simulation, leading to staggered startup periods of hours of simulated time, while the actual satellite overflight periods and therefore the measurement time that we are interested in are in the order of minutes per satellite.

Roughly speaking, CAMELAMA's bootstrapping works as follows:

- Before entering the regular state, where the CAMELAMA protocol mechanism controls medium access, a node enters an *introductory state* where it sends only *introductory messages*.
- The payload of introductory messages is identical to regular messages, except that they are flagged as introductory using a special value $v = 2^{16} - 1$. They are interpreted as $v = 0$ and as requesting an acknowledgment from the nearest regular-state node.
- Introductory messages are sent probabilistically with a small probability p_{tx} and in an anti-LAMA pattern with respect to *fire positions*, i. e., only if the node's nearest fire position is *more* than $d_{\text{min}}^{\text{intro}}$ away. $d_{\text{min}}^{\text{intro}}$ is a threshold parameter that we set to $L/2$.
- A node transitions from introductory state to regular state if it receives an ACK message, meaning that its nearest regular-state neighbor node has successfully received one introductory message.
- If a regular node is requested to transmit ACK messages, it uses a share of *its* slots to transmit ACK messages, i. e., a share of the slots that are assigned to it by the regular CAMELAMA mechanism. An ACK message is simply bit-by-bit the exact same message it acknowledges, except for the field v whose value is replaced with $v = 2^{16} - 2$ to flag it as an acknowledgment.
- Nodes in regular state do not perform any probabilistic listening.

The idea behind this is simple: in introductory state, nodes transmit only in slots where the nearest transmitting regular node is most likely far away. Therefore, an introductory message is received by the nearest neighbors with a high probability. To avoid multiple

nearby introductory-state nodes to transmit simultaneously, they "play slotted ALOHA" on top, i. e., even in slots where the nearest fire position is sufficiently far away, a node draws a pseudorandom number to decide whether to send or not.

The ACK mechanism ensures that an introductory-state node switches to regular state only if at least one of its messages was received by its nearest neighbor. LAMA's protocol state machine is shown in Fig. 3.4. If we required every node to receive an ACK message from a regular-state node to enter the regular state, no node could ever be the first one to do that. We therefore added second path to regular state, where nodes without a regular-state neighbor within their base send region enter regular state after a fixed number ($N_{\text{alone}}^{\text{tx,max}}$) of transmissions. Considering only nodes within the base send region here is justified by the following argument: if there is no regular-state node inside the base send region, then a neighborhood of non-vanishing area around the node's position is outside every regular-state node's base send region, meaning that no nearby regular-state node may transmit if the **np** falls into that area.

This bootstrapping algorithm takes three new parameters:

1. $d_{\text{min}}^{\text{intro}}$, the minimum distance from the fire position needed to transmit an introductory message. In the perspective of a certain node, we term such slots *introductory slots*.

2. p_{tx}, the transmission probability in introductory state.

3. $N_{\text{alone}}^{\text{tx,max}}$, the number of introductory messages need to be sent for nodes that are *alone*.

Compared with the single parameter that probabilistic listening requires, this seems to worsen the situation in terms of protocol complexity. But these parameters affect the bootstrapping phase only, in contrast to P^{a}, which also affects regular operation in the long term.

With a large distance $d_{\text{min}}^{\text{intro}}$, the anti-LAMA condition $\text{dist}(\text{self}, \mathbf{np}) > d_{\text{min}}^{\text{intro}}$ ensures that the nearest fire position, i. e., the position where the nearest regularly transmitting node is probably located, is sufficiently far away and should therefore be as large as possible. If it is too large, however, the area fraction of the sphere that is more than $d_{\text{min}}^{\text{intro}}$ apart from any fire position is small or even vanishes. We therefore set it to the most obvious choice of $L/2$ which is equal to the largest radius for which open spheres around every fire position do not intersect. With our Tammes-set coverage around 84% approximately every 7th slot is on average a candidate for introductory messages. If we choose L equal to or larger than twice the typical 50%-rx-probability range, an introductory-state node transmits only in slots where the nearest sender is likely to be beyond this range.

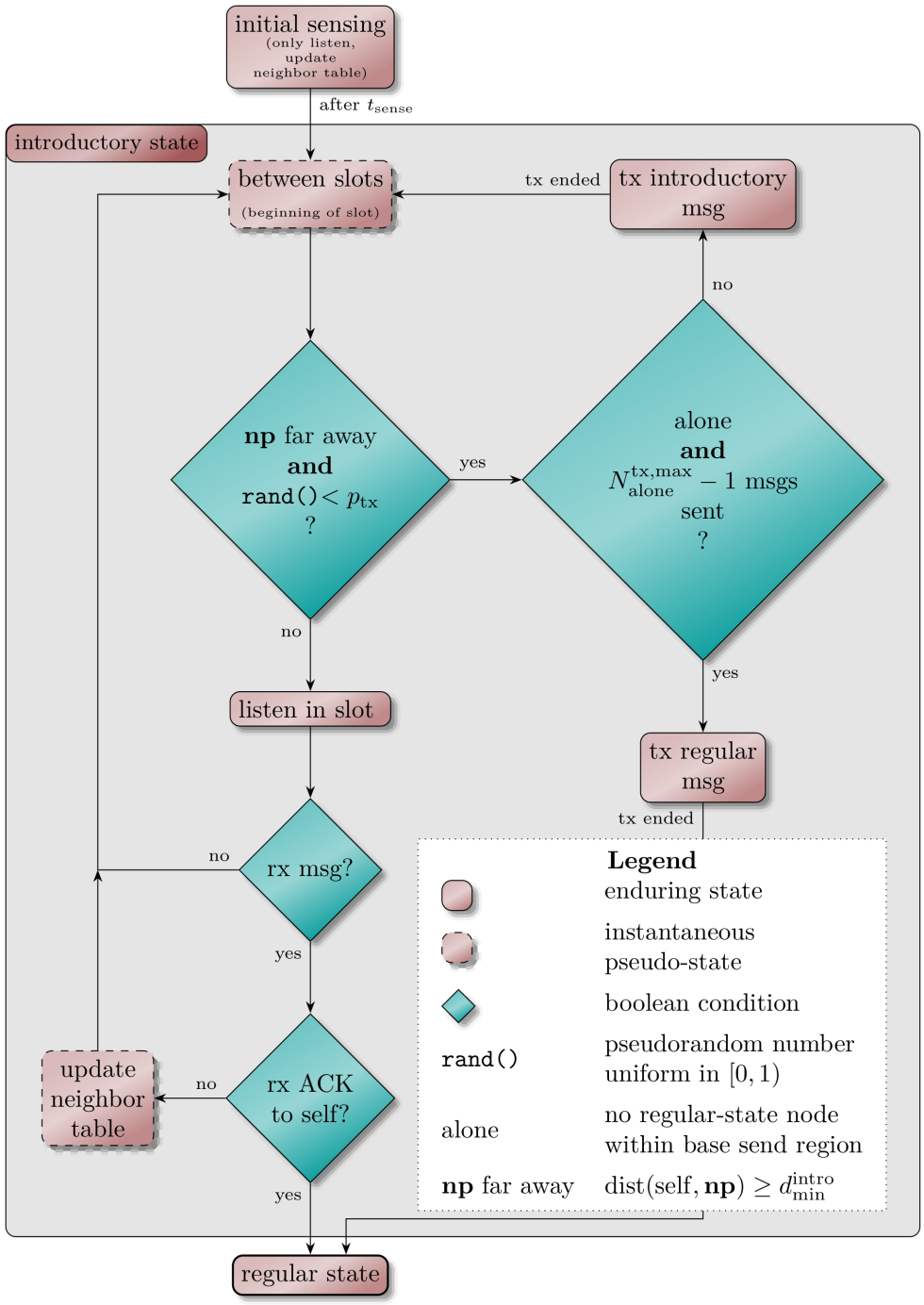


Fig. 3.4.: State machine of CAMELAMA's introductory state used for robust bootstrapping.

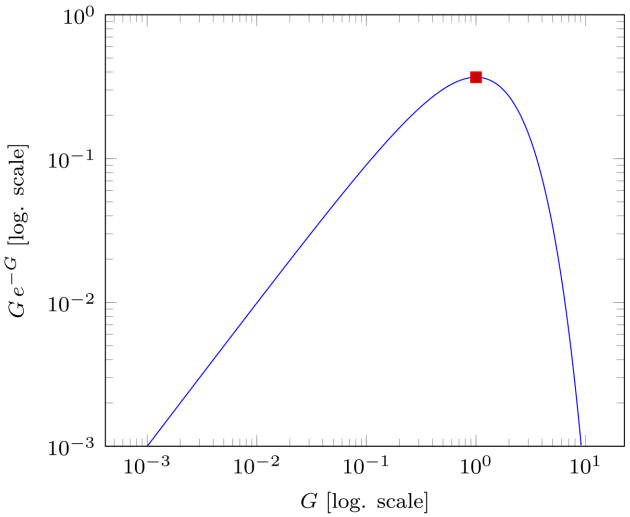


Fig. 3.5.: Plot of $G\,e^{-G}$. Choosing p_{tx} too small, implying G below its optimal value, leads to a linear decrease of channel utilization whereas choosing p_{tx} too high leads to an exponential drop-off.

The number $N_{\text{alone}}^{\text{tx,max}}$ of introductory messages that need to be sent for ACK-less initialization was set to 3 by us. We did not try different values because 3 worked fine and because this parameter only affects the very first few of nodes switching to regular mode anyway.

The transmission probability p_{tx} for introductory messages is the only parameter that lacks an obvious or well-justified value to choose. Intuitively one can assume that it should be adapted to approximately the number of nearby simultaneously introducing nodes, but this number is highly scenario-dependent and therefore cannot be a property of the protocol. When we consider a group of N nodes mutually less than ϵ apart, where $\epsilon \ll L$, so that the anti-LAMA condition is roughly equivalent for all N nodes, the per-slot probability that exactly one of these nodes transmits at a time is given by (3.9), a plot of which is shown in Fig. 3.5.

$$\begin{aligned} P(1 \text{ out of } N \text{ nodes transmitting}) &= N p_{\text{tx}} (1-p_{\text{tx}})^{N-1} \\ &= \frac{G}{1-p_{\text{tx}}} \left(1 - \frac{G}{N}\right)^N \qquad \text{with } G := p_{\text{tx}} N \\ &\xrightarrow[G \text{ const}]{N\to\infty} G \exp(-G) \end{aligned} \tag{3.9}$$

As this probability $G\,e^{-G}$ is maximized for $G = 1$, i. e. $p_{\text{tx}} = N^{-1}$, and decreases linearly for $N \ll \frac{1}{p_{\text{tx}}}$ but exponentially for $N \gg \frac{1}{p_{\text{tx}}}$, it is favorable to overprovision by choosing $\frac{1}{p_{\text{tx}}}$ to be much higher than the highest expected numbers of N. For a given node-density ρ, the number of nodes contending for the same set of anti-LAMA-slots is upper bounded by

$\rho\frac{\pi L^2}{4}$. The densest scenarios we considered in our evaluation are 1600 nodes in $14\,400\,\mathrm{NM}^2$. For our experiments we chose $p_{\mathrm{tx}} = 10^{-2}$ which worked fine.

Since the simultaneous start of all protocol nodes is rather an artifact of our simulation-based experiments than a common situation in practice, we abstained from quantitative evaluation of the bootstrapping algorithm's performance. Instead, we substituted probabilistic listening with the new bootstrap algorithm and adapted the time-framing in our evaluation: we started simulations without staggering node startup and dynamically adjusted the equilibration time that was excluded from measurements to end two minutes after the last node reached regular state. Once all nodes in a simulation are in regular state, introductory messages are never sent anymore and the whole bootstrapping mechanism becomes effectless for the rest of the simulation. The mere fact that this mechanism successfully bootstrapped the protocol in each simulation that we conducted, ipso facto is a strong empirical argument for it to be sufficient for its very purpose.

A Note on Loss of Mutual Awareness We suppose that even after bootstrapping, when every node is in regular state and aware of its nearby nodes, there is a small but finite probability that this state of mutual awareness deteriorates due to packet losses. This probability may also be increased by maliciously behaving nodes that jam the channel or even transmit spoofed packets in order to evoke a denial of service.

For reasons discussed in Section 3.2.2 the LAMA protocol is not able to recover from such a loss of mutual awareness. We have briefly verified this assumption using a small jamming experiment (simulation), where we applied a jam signal until the neighbor tables of all nodes where emptied.

Security considerations in general and denial-of-service (DOS) mitigation in particular are beyond the scope of this work. Nevertheless, we want to state some ideas how this problem could be tackled:

1. So far we assumed that from a protocol perspective, a LAMA controller can only receive a message in a slot or not. If it listens in a slot and no packet is received, it cannot tell the difference between no node on Earth transmitting and a hundred nodes all 10 m away transmitting simultaneously. Modern receivers, however, are able to report a received signal strength indicator (RSSI) to the host system which can then be used to discriminate message collisions of two or more nearby nodes from messages transmitted too far away. LAMA nodes could then fall back to introductory mode if a certain threshold of the average number of slots with nearby-collisions is exceeded.

2. Even without RSSI, a breakdown of mutual awareness could be detected by inspecting the neighbor table only. An empty or near-empty neighbor table means that either the mutual awareness has deteriorated and practically every packet collides with nearby messages or the node is literally *alone* in its area. In the former case, falling back to introductory state would help the system of all nodes recover to a mutually aware state. In the latter case, medium access is practically trivial as there are no potential receivers anyway.

We re-emphasize that these are only ideas to mitigate a hypothetical problem. We neither implemented nor tested these ideas, because in our evaluation we were never confronted with the problem of deteriorating mutual awareness of nearby nodes.

3.2.3 Terrestrial Cooperative Awareness and Orbital Overhearing

In case of vessel safety at the high seas, broadcasting navigational data serves both the primary goal of pure terrestrial cooperative awareness, and the secondary goal of global surveillance through overhearing by LEO satellites. We have already discussed that good spatial reuse and therefore high channel utilization for terrestrial reception can be achieved when the distance between nearest simultaneous senders is approximately twice the typical maximum of terrestrial transmission range. Unfortunately, this pattern of spatial reuse is counter-productive for orbital overhearing, if it generally leads to multiply nodes transmitting simultaneously within the satellite's receiving antenna main lobe (see Fig. 3.0). To solve this problem, we use two base fire positions sets, one with a small value of L that is optimized for terrestrial transmissions and the other with a much larger parameter L for orbital overhearing. These base fire position sets are applied in a simple fixed interleaving pattern: given an integer protocol parameter N_{orb}, the finely granular base fire position set is used in *terrestrial slots* $i \equiv 0 \pmod{N_{\text{orb}} + 1}$ (where i is the integer slot number) while the coarsely granular base fire position set is used in all other slots (from here on called *orbital slots*). Bootstrapping packets, i. e., introductory and acknowledgment messages, are restricted to terrestrial slots only. Achieving channel access fairness in orbital slots usually requires much larger values of v than in terrestrial slots, because if fewer nodes can transmit simultaneously then each node transmits less frequently. In order to handle this appropriately, each vow of silence applies only to the type of slots it was transmitted in. In our implementation, each node does not only count up total slot numbers in order to generate pseudorandom fire positions and determine the slot type, but also counts up per-type slot numbers i_{ter} and i_{orb} satisfying $i_{\text{ter}} + i_{\text{orb}} = i$ that are used only for the vow-of-silence fairness mechanism. For a visual overview, see Fig. 3.6.

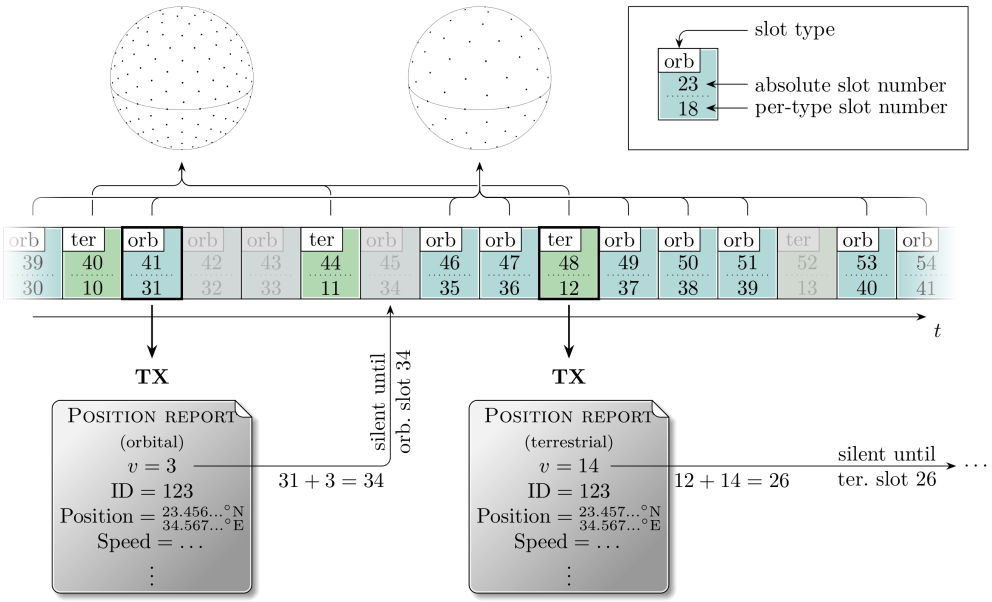


Fig. 3.6.: Interleaving of slot types (terrestrial/orbital) in CAMELAMA. In terrestrial slots, a base-fire-position set with L smaller than in orbital slots is used. The slot types and -numbers are hardcoded into the protocol and therefore synchronized for all nodes. In addition, an example of the semantics of the vow-of-silence field v of two transmissions of a hypothetical node "123" is given. v refers to per-type slot-numbers and affects slot-assignment only within the same slot type. Slots in which node 123 cannot send due to the vow-of-silence mechanism are greyed out. Note that the base-fire-position sets shown in the figure are both drawn with L much greater than in the evaluation. The slot numbers and slot types are not part of a transmission's payload as they can be inferred from the time of reception.

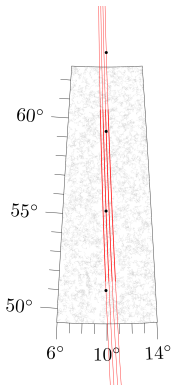


Fig. 3.7.: Vessel mobility and satellite ground tracks of a 1500 nodes run.

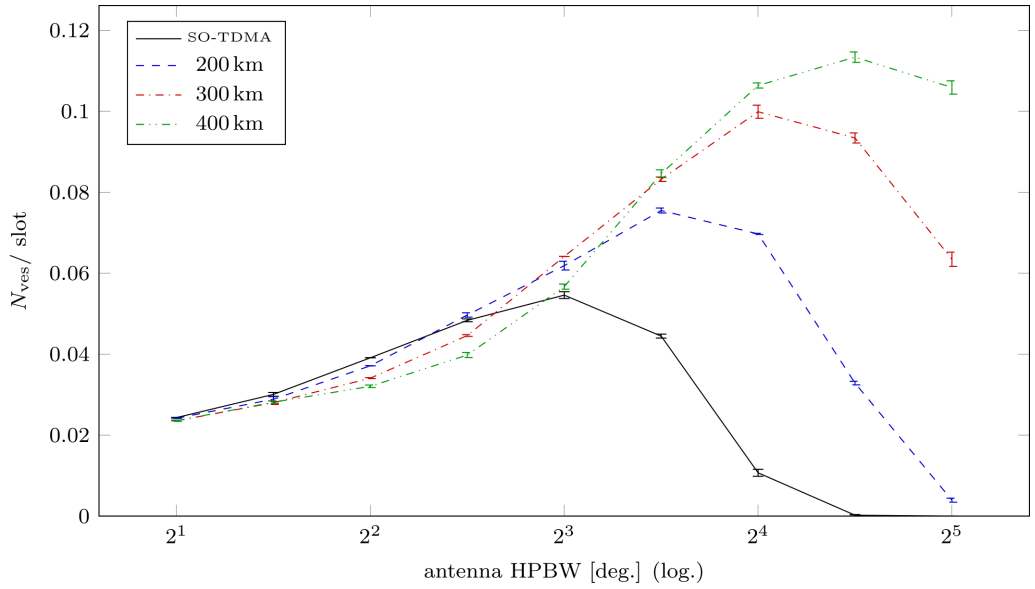
3.3 Evaluation

We evaluate CAMELAMA using ns-3 [128] in a setting motivated by AIS and compare it against SO-TDMA. Our primary focus lies on the quantitative analysis of the successful reception of vessels' position reports by LEO satellites.

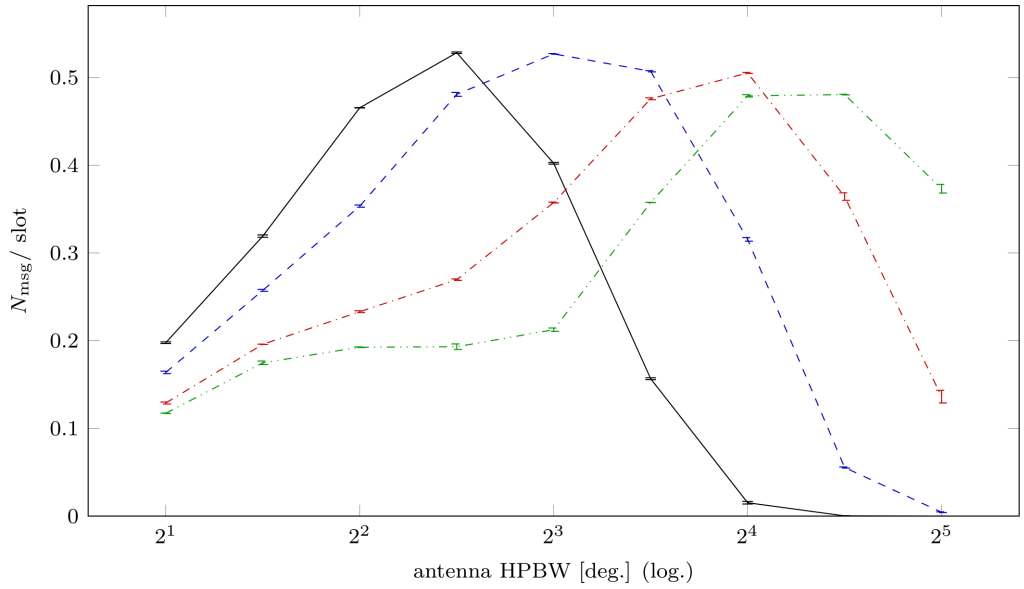
3.3.0 Mobility

Ideally we would have liked to evaluate the protocol with a global Earth-scale configuration of vessels, complemented by an Earth-covering constellation of monitoring satellites as proposed in [34]. To produce trustworthy results we decided to model transmission errors based on the signal to interference plus noise ratio. In ns-3, using a linear list for range searching, this implies computation cost per transmission to be linear in the global number of nodes. This in turn leads to per-simulation computational cost quadratic in the number of nodes if the nodes' average transmission rate is kept constant. Given that our largest simulations of 6000 nodes already take several hours per run, simulations with realistic global ship counts (400,000–550,000 nodes as of 2016 [35]) would be prohibitively expensive. Improved range searching techniques based on grids and kd-trees [54] rely on in-euclidean-plane mobility and are therefore not applicable to Earth-spanning node distributions.

We therefore decided to use scenarios where the simulated vessels are confined to a rectangular box in latitude/longitude parametrization. We modeled vessel mobility using a spherical waypoint mobility model, i. e., each vessel's mobility is described in terms of a sequence of waypoints (i. e., time-position pairs); the node moves with constant speed along the shortest path between consecutive waypoints.



(a) Discovered vessels per time slot.



(b) Message count per time slot.

Fig. 3.8.: Number of (a) discovered ships N_{ves} and (b) received messages N_{msg}, each per slot per satellite, collected with antennas of varying beam width, during one formation overflight of a 3000 nodes random walk topology performing SO-TDMA or CAMELAMA, the latter with $L_{\text{orb}} \in \{200, 300, 400\}$ km and $N_{\text{orb}} = 4$. Legend of (a) applies to (b) as well.

For most experiments we used a random walk mobility [131] with speed drawn UAR from $[0, 30\,\frac{\mathrm{m}}{\mathrm{s}}]$, inter-waypoint times are drawn UAR from $[0, 300\,\mathrm{s}]$, and initial positions drawn UAR with respect to areal probability density from the confining rectangle, i. e., $\sin\theta$ uniform in $[\arcsin\theta_{\min}, \arcsin\theta_{\max})$. We verified experimentally that this mobility model has a steady state and that it is initialized in this steady state to avoid pitfalls like [55]. In some experiments we used real-world AIS traces captured in the area of Denmark at 1:00pm–1:30pm (UTC) on 2017/06/01.[3] The traces were limited to those nodes corresponding to AIS Class A devices with at least one waypoint inside the rectangular region of latitude $[49.26°\mathrm{N}, 62.74°\mathrm{N}]$ and longitude $[5.97°\mathrm{E}, 14.03°\mathrm{E}]$, i. e., a rectangle centered at 56°N, 10°E with a latitude extent of $1500\,\mathrm{km}$ and a longitude extent of $500\,\mathrm{km}$ at 56°N. This results in 1688 nodes. The random walk topologies were generated for the same rectangle.

In order to study message reception of multiple satellites passing over the same region, we put four satellites in a string-of-pearls orbital configuration with $500\,\mathrm{km}$ altitude, vanishing eccentricity, 90° inclination, and $500\,\mathrm{km}$ inter-satellite distance. The remaining orbital elements were adjusted such that the formation's geometrical center passes over the rectangle's center point and such that the first satellite's footprint enters the rectangle just after we consider the terrestrial MAC protocol as equilibrated. The satellites' mobility was then modeled with SGP4 and converted to Earth-centered, Earth-fixed (ECEF) Cartesian coordinates used for vessel mobility.

3.3.1 Channel Model

We implemented CAMELAMA and SO-TDMA for a single $25\,\mathrm{kHz}$ bandwidth VHF channel at $161.975\,\mathrm{MHz}$ center frequency modulated in binary GMSK with $9600\,\mathrm{bit/s}$. Slots of $\frac{2}{75}\mathrm{s} \mathrel{\widehat{=}} 256\,\mathrm{bit}$ were used. The signal-to-interference-plus-noise based packet loss model and the path loss of terrestrial ship-to-ship communication was taken from [3]. Path loss computation for ship-to-satellite signal propagation was ported from the ESTNeT simulator [56].

3.3.2 Antenna Model

A $500\,\mathrm{km}$-altitude satellite with an omnidirectional antenna has a swath width of $\approx 5000\,\mathrm{km}$ horizon to horizon. We can neither simulate realistic-node-density topologies of this size (Section 3.3.0), nor do we have access to corresponding captured AIS traces. Thus, we limit the swath width with an abstract model of a nadir-pointing directional antenna.

[3] `ftp://ftp.ais.dk/ais_data/dk_csv_jun2017.rar`, accessed 12/02/18.

It is parametrized by its half-power beam width (HPBW) and has a directional gain (3.10) depending only on the angle α between the received signal and the nadir direction.

$$A(\alpha) = c \cdot 2^{-\left(\frac{2\alpha}{\mathrm{HPBW}}\right)^2} \qquad c \in \mathbb{R} \colon \int A \, \mathrm{d}\Omega = 4\pi \tag{3.10}$$

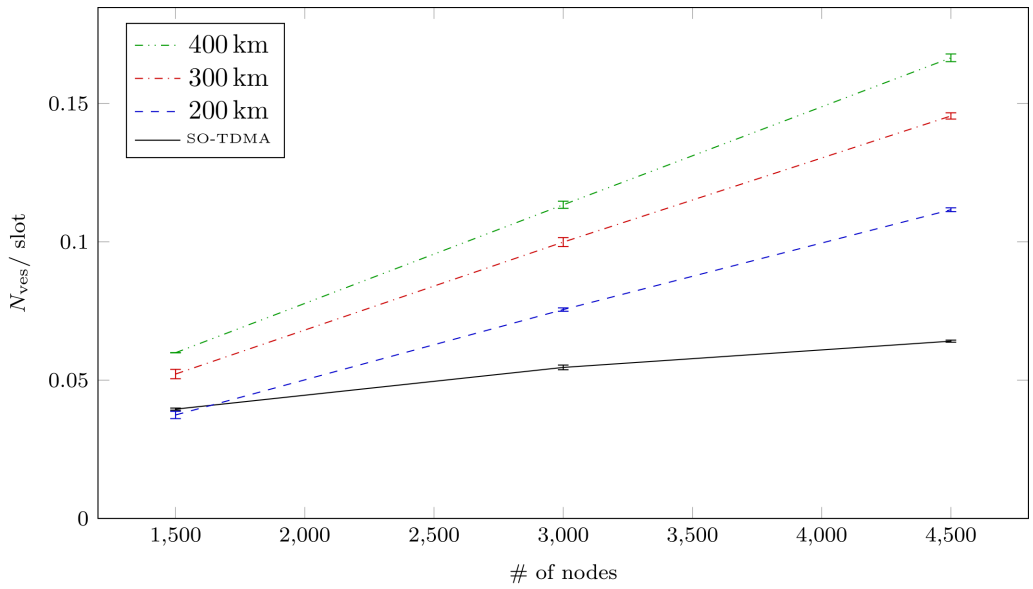
This abstract model allows evaluating protocol performance depending on HPBW regardless of whether the receiver uses a physically directive antenna, digital beamforming using a patch antenna array, or creates the effective FoV by means of signal processing using timing- and Doppler-shift-based filtering on the received signals.

3.3.3 Split Simulation Strategy

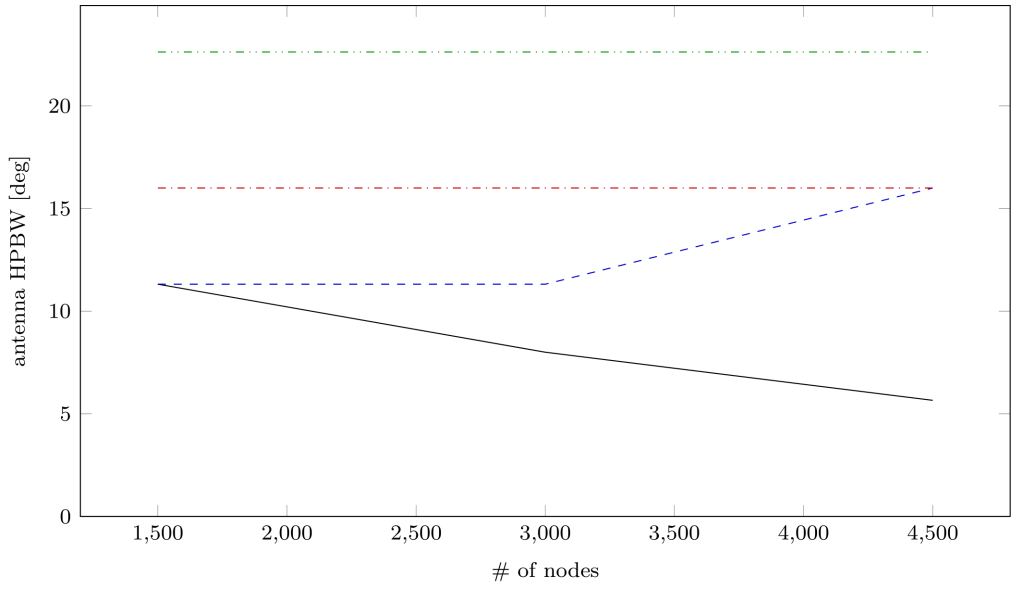
As our protocol contains no transmissions from satellites at all, the vessels' behavior is completely independent of the satellites' behavior. We use this to split each simulation into two separate parts: 1) a terrestrial-only simulation of CAMELAMA or SO-TDMA where we record an exact schedule of which message is transmitted in which slot, and 2) a satellite-reception-only simulation, where the vessels replay the transmission schedules recorded in the first simulation part while the vessels' and satellites' mobility are simulated. The second simulation part is much cheaper, because the vessels' behavior is fixed and independent of the messages that are received. Therefore, the loop in ns-3 that iterates over all nodes in the channel to compute signal-power levels for every node for every transmission can be limited to the satellites only, avoiding the need to perform this computation for thousands of terrestrial nodes. We leveraged this computational simplification by re-using the same terrestrial simulation part for different satellite configurations and receiver-antenna configurations that we examined. Unless stated otherwise, each data point corresponds to six independently seeded simulations and error bars in plots depict the standard error. Distances given as legend keys denote L_{orb} values.

3.3.4 Measurement Timing

We conducted measurements with different antenna HPBWs with the widest swaths just smaller than the longitudinal width of the terrestrial topology. If we measured data in situations with only part of the terrestrial topology in the satellite's FoV, boundary effects would not only affect our results, but their magnitude would vary with the antenna's HPBW, making it hard to tell apart boundary effects from the desired underlying performance characteristics. To circumvent this problem we seek to essentially cut out the boundary effects by limiting the measurement of in-orbit reception per satellite to the time span when the satellite's ground track is within the terrestrial topology's latitude range shrunk



(a) # of vessels discovered.



(b) Optimal antenna beam width.

Fig. 3.9.: (a) N_{ves} per slot for per-parameter-optimal antenna beam width (corresponding to the curve maxima in Fig. 3.8) and (b) the corresponding beam widths. Error bars are omitted for (b) because the values shown correspond to an argmax out of a finite set of parameter values. The legend of (a) applies to both plots.

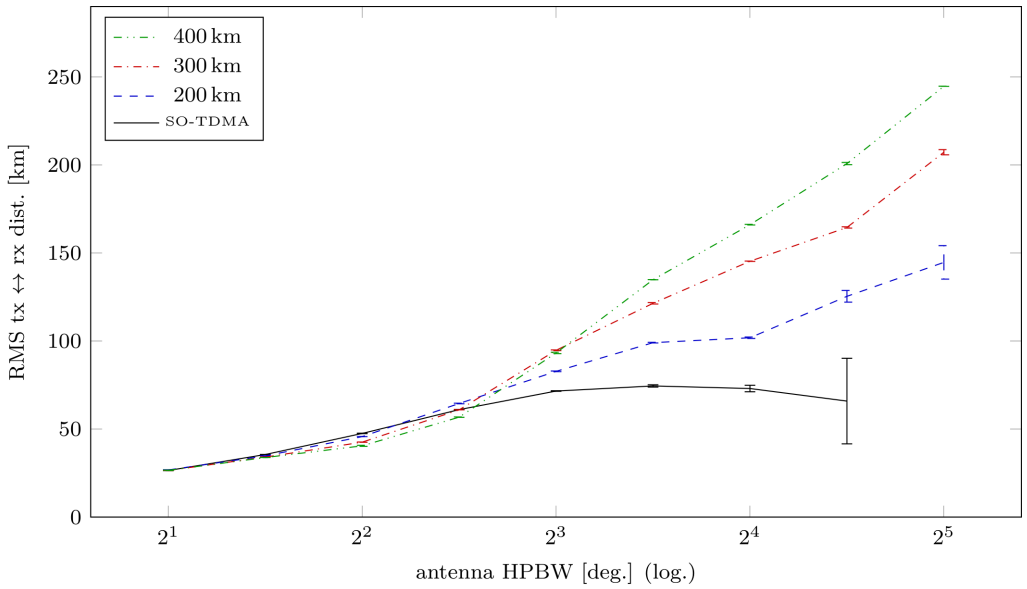


Fig. 3.10.: RMS distance from sending ship to receiving satellite's ground track measured in runs depicted in Fig. 3.8a,b.

by 250 km. This way we can assure that the satellite's ground track during measurement is surrounded by $\gtrapprox 200$ km of populated ground area (see Fig. 3.7) in every direction.

Every measurement of SO-TDMA scenarios is preceded by 10 min of equilibration time to allow SO-TDMA to reach a steady state. Every measurement of CAMELAMA scenarios is preceded by equilibration time consisting of time needed for all nodes to reach regular state plus 2 min, which, when taken together, was less than 10 min in total in each simulation run.

3.3.5 Parameters

In the evaluation performed in [3], the LAMA protocol proved robust with respect to its parameters. CAMELAMA is based on LAMA, so in this evaluation we used $L_{\text{ter}} = 92.6\,\text{km} (= 50\,\text{NM})$, $r_{\text{max}} = 23.15\,\text{km} (= 12.5\,\text{NM})$, $\nu = 0.8$, and $N_{\text{orb}} = 4$ (see Section 3.3.7) together with $L_{\text{orb}} \in \{200, 300, 400\}$km and antenna beam widths $HPBW \in [2^\circ, 32^\circ]$.

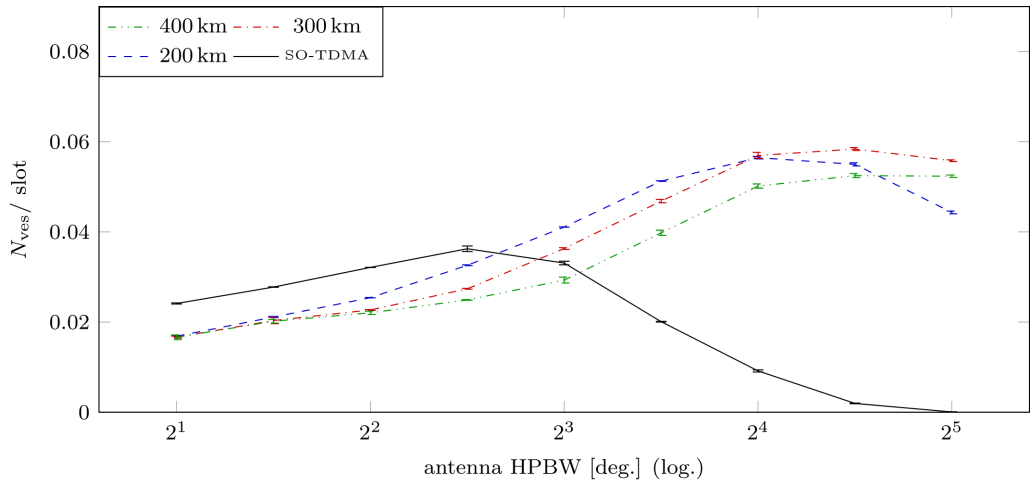
3.3.6 Satellite-Based Monitoring

When a satellite formation passes over the simulated area of Earth, there is some redundancy in the received messages. The same message can be received by two or more satellites at the same time, and multiple different messages of the same vessel can be

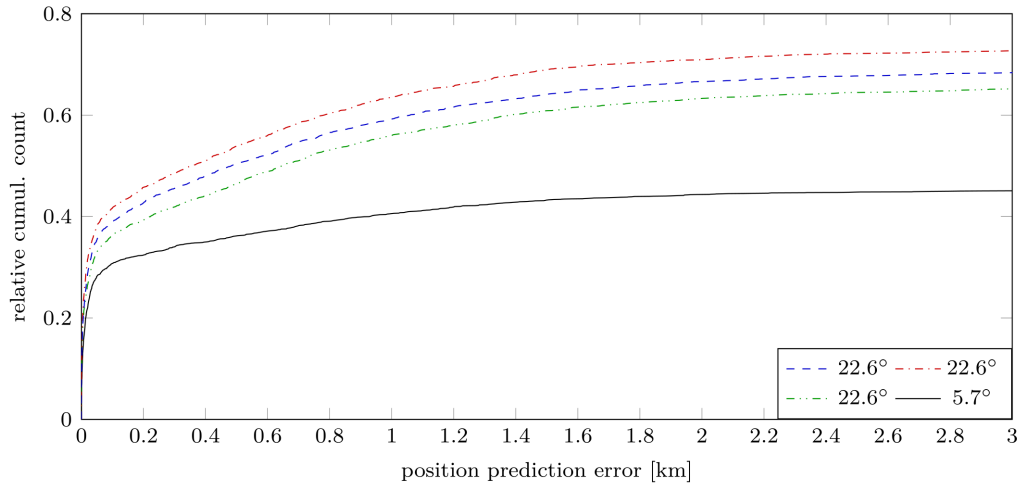
received, carrying different but highly correlated information. We consider both, the number of messages received N_{msg} and, deduplicating messages from the same sender, the number N_{ves} of "discovered" vessels, i. e., vessels that at least one message is received from. These total counts were then divided by the sum of time slots regarded for reception over all four satellites. Fig. 3.8 shows both metrics over antenna HPBW for SO-TDMA and for CAMELAMA with $L_{\text{orb}} \in \{200, 300, 400\}$ km measured for 3000 nodes random walk topologies. Message counts received from SO-TDMA are on par with CAMELAMA. However, the former requires significantly higher antenna gain which is unfavorable as discussed in Section 3.2.0, while the latter leads to more distinct vessels being discovered. As expected, the optimal antenna HPBW, i. e. the curves' arg maxima, increase with L_{orb}. A corresponding measurement based on real AIS traces (Fig. 3.11a) shows that CAMELAMA still performs better but the strong correlation of antenna beam width and L_{orb} vanishes.

We repeated this measurement for random walk topologies of identical geometrical bounds but with varying node counts. Fig. 3.9a shows N_{ves} per slot over node count where each data point was measured for the optimal HPBW. The corresponding optimal HPBW values are given in Fig. 3.9b. For CAMELAMA, the number of vessels detected increases while the optimal beam width shows no significant dependence on node density. SO-TDMA shows a significantly weaker increase along with a narrowing of the optimal antenna directiveness, indicating that receiving SO-TDMA signals at high node densities requires high-gain antennas to compensate for smaller distances between simultaneous senders.

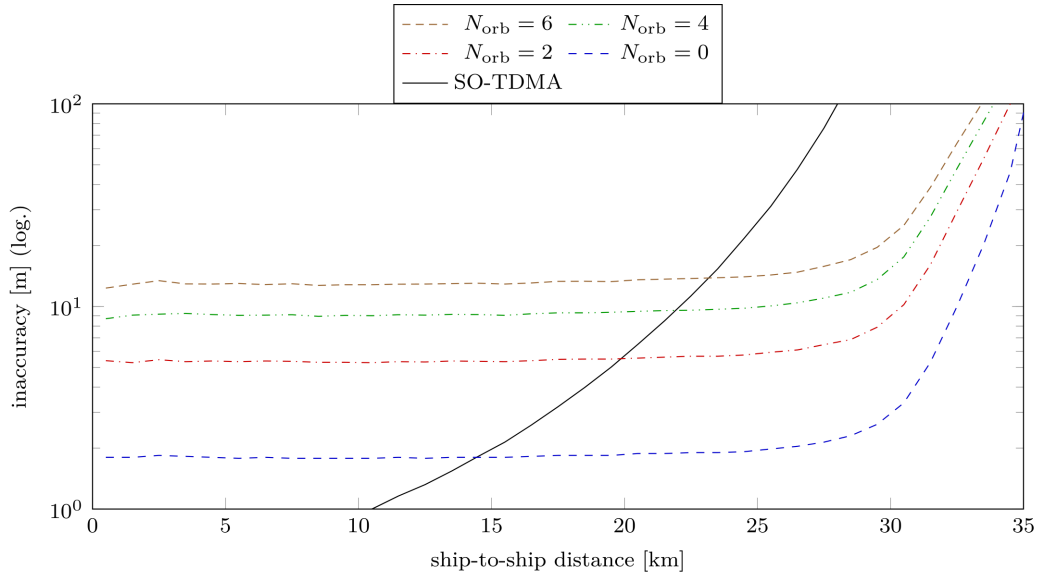
For every received message, we also measured the distance of the transmitting node to the receiving satellite's ground track point to get a sense of the "effective swath width" that a satellite is receiving messages from. In Fig. 3.10 these RMS distances that correspond to the data points in Fig. 3.8 are shown. CAMELAMA effectively receives messages from a significantly larger FoV that increases roughly linearly with growing value of L_{orb}, just as one would expect. Finally, we measured the monitoring accuracy in the scenario of real traces at the per-MAC-protocol-optimal HPBWs. At the end of the satellite reception measurement time interval, we recorded for each vessel in the simulation the distance between its true position and the position resulting from extrapolating the last beacon received in orbit to this point in time using dead reckoning. For vessels no beacon at all was received from, ∞ was recorded. Fig. 3.11b shows the relative cumulative counts and reveals that while CAMELAMA achieves a tracking accuracy smaller than 1 km for 55%–64% of vessels, SO-TDMA provides the same accuracy only for 40%.



(a) N_{ves} vs. HPBW for real AIS traces.



(b) Orbital tracking inaccuracy.



(c) Median terr. location-prediction inaccuracy.

Fig. 3.11.: (a) N_{ves} per slot over sat. HPBW for real vessel traces (legend denotes L_{orb}) and (b) a CDF-plot of the corresponding tracking (in-) accuracy measured at $5.7°$ (SO-TDMA) and $22.6°$ (CAMELAMA) HPBW. Legend of (a) applies to (b) as well. (c) Median of terrestrial location-prediction inaccuracy over ship-to-ship distance in random walk mobility for SO-TDMA and CAMELAMA at $L_{\text{orb}} = 400\,\text{km}$.

3.3.7 Terrestrial Performance

We have re-run the whole evaluation from [3] for CAMELAMA with $N_{\text{orb}} = 0$ (i. e., without orbital slots at all) and found that CAMELAMA without orbital slots behaves qualitatively equivalent to LAMA, outperforming SO-TDMA. To quantify the effect of orbital slots on the performance of terrestrial cooperative awareness, we measured the neighbor location prediction inaccuracy (the distance between neighboring nodes' true positions and the predicted positions extrapolated from the last received position report's location, speed, and course) depending on N_{orb}, using the methodology of [3], Section 4.3. In a $(222.24\,\text{km})^2$ (120 NM) square topology of 1600 random-walk nodes we measured once every 7 s the location-prediction inaccuracy of every ordered node pair and the node pair's true distance. We then binned these samples by the node distance. In Fig. 3.11c the prediction inaccuracy median is plotted over the ship-to-ship distance for SO-TDMA and CAMELAMA with varying $N_{\text{orb}} \in \{0, 2, 4, 6\}$. As we take away channel resources in terms of terrestrial slots, the inaccuracy increases from less than 2 m to more than 10 m, which is still below a typical vessel size. Nevertheless, CAMELAMA achieves to maintain a nearly constant accuracy up to distances of 30 km.

3.4 Conclusion

We introduced CAMELAMA, a location-assisted MAC protocol for cooperative awareness beaconing combined with satellite-based traffic monitoring. We demonstrated that it leads to significantly higher node discovery rates and tracking accuracy compared to SO-TDMA. This already holds for a simple static nadir-pointing satellite antenna and a receiver naïvely treating interference as noise in decoding. Since improving the MAC protocol is orthogonal to recent efforts to improve in-orbit de-collision of AIS messages, we see great potential in combining these techniques. If nodes use CAMELAMA, a satellite knows approximately where transmitting nodes are located. This knowledge could be exploited with respect to Doppler shift and propagation delay when using techniques as in [36] as well as for direction-dependent antenna gain adaption, i. e., receive antenna beamforming [41]. As SO-TDMA lacks this transmitter-location knowledge, the advantages of CAMELAMA might further increase significantly if improved decoding techniques are applied. As this chapter is merely a proof of concept, consideration of environmental effects like adverse atmospheric conditions is left for future work.

4 Opportunistic In-Orbit Forwarding and Aggregation

> *The future is already here — it's just not very evenly distributed.*
>
> — **William Ford Gibson**
> (American-Canadian writer)

This chapter is largely based on our own publication *"Evaluation of a delay tolerant networking approach for inter-satellite communication in LEO for time sensitive traffc monitoring"*[0] published as a full paper at the 2015 IAA Symposium on Small Satellites for Earth Observation.

TL;DR *Simple probabilistic in-orbit forwarding and aggregation of AIS messages can effectively reduce monitoring delays in Earth-spanning satellite constellations used for vessel traffic monitoring.*

4.0 Introduction

As we have discussed extensively in the last chapter, beacon messages of systems such as AIS or ADS–B can be overheard by spacecraft in low Earth orbit. To achieve the goal of satellite-enabled global vessel traffic monitoring, however, the navigational data that is received in orbit must somehow make its way to a satellite ground station to be processed and used, e. g., in data centers. When using only a small number of GSs, a satellite will for most of the time (also discussed in Chapter 7) not be in direct communication range to any GS. In this chapter we propose to make use of ISLs to forward the data contained in the received beacon messages between satellites to decrease the end-to-end delay of navigational data from the originally transmitting vessel to the satellite GS network. Considering only a medium number of 18 nano-satellites, even with ISLs there is no end-to-end multi-hop connection between each satellite and the GS network for most of the time. In this chapter we propose a store-and-forward approach that is robust with respect to noise and loss of individual satellites, yet simple and scalable. It follows the delay-tolerant networking paradigm and relies on payload forwarding through stateless

stochastic broadcasting. Unfortunately, it did not turn out feasible to integrate Chapter 3 and this chapter into one evaluation scenario: forwarding of vessels' navigational data over ISLs has very limited effect in small satellite groups in dense orbital configuration that were considered in Chapter 3. Both the global Walker constellation and the multiple-hour measurement periods considered in this chapter, however, are not simulatable with reasonable effort using the methodology of Section 3.3, as we discussed in Section 3.3.0.

The remainder of this chapter is structured as follows. In Section 4.1 we outline the overall design of the proposed satellite system. We subsequently go into more detail specifically on the protocol side in Section 4.2. In Section 4.3 we describe the evaluation setting and present the empirical results. We finally conclude with a summary in Section 4.4.

4.1 System Design

The proposed message forwarding approach is designed for observation of any types of static or moving ground vehicles. For the evaluations presented here, the specific case of ship tracking is considered. The proposed tracking system leverages the existing AIS, which is used by vessels to periodically transmit ID and position data in radio beacons. AIS messages are sent periodically by all ships equipped with AIS transmitters. AIS is based on SO-TDMA where each message is broadcasted in a distinct $\frac{1}{2250}\text{min}$ time slot in one of two VHF maritime mobile channels. Within an AIS-message, the sending ship pre-announces the time slot for its next transmission, thereby providing a mechanism to avoid collisions of AIS-messages of nearby ships. The AIS beacons are captured by a distributed small satellite system. Without inter-satellite communication, the minimum total monitoring delay is the time it takes the observing satellite to come into view of a ground station after observing the vessel. Shorter times can be achieved if the satellites forward collected data between each other. To reduce the amount of unnecessary information and to make the best use of available capacities in a satellite-based AIS monitoring system, only relevant information should be forwarded to users on the ground. To this end, the received AIS messages are processed in the satellite network. The data forwarding approach that is pursued here is motivated by the following consideration: for continuous monitoring of current vessel positions, it is not required to maintain old data in the system if more up-to-date information for the same vessel is already available. Thus, if a satellite holds information of a vessel with a certain time stamp, and receives another data set of the same vessel with a different time stamp—either directly from the vessel or from another satellite—then it needs to keep only the most recent data. Our proposed message forwarding approach makes use of this consideration in the following way: regularly, each satellite transmits a random subset of locally available data. Upon reception of such a message by another satellite,

the received information is merged with the locally available knowledge according to the policy of always keeping the most recent information per vessel. This protocol makes use of all possible paths in the network stochastically in parallel, which yields inherent robustness with respect to network disturbances, including transmission errors or the loss of individual satellites, while at the same time not requiring any explicit coordination or path planning.

4.2 Message Forwarding

The strategy of maintaining only the most recent data per vessel implies that the number of vessel data sets in any satellite's send buffer is always bounded by the total number of AIS-equipped vessels on earth. We restrict our protocol to receiving class A AIS position reports, which consist of 168 bit of payload data. An AIS record in the satellite network consists of 141 bit from the original AIS position report (everything apart from *Message ID*, *Repeat indicator*, and *Communication state*) and a 17 bit reception time stamp holding the seconds elapsed since last midnight (UTC) as an unsigned integer. This results in a record size of 158 bit. To avoid ambiguity of the reception time stamp, records older than 23 h are removed from the database. For medium access we use pure ALOHA random access for sending frames via the omnidirectional satellite antenna. A frame consists of a fixed-length payload along with a checksum. The frame size and the (also fixed) transmission data rate determine the time T needed to transmit a frame on the channel. When a satellite finishes a frame transmission, it chooses the time it waits before the start of the next transmission randomly with an exponential distribution with mean t_{wait}. For our simulation, we chose $t_{\text{wait}} = 3T$. This value is a compromise between the theoretically optimal ALOHA inter-frame times for the situations where a satellite is in range of either one or two other satellites. Before transmitting a frame, the payload data is constructed by choosing a random subset of appropriate size from the set of all AIS records in the satellite's local database. When a satellite successfully receives a frame (i. e., if it receives a complete frame and positively validates its integrity using the checksum), it updates its database by keeping only the most recent available data set for each vessel on which information was contained in the frame.

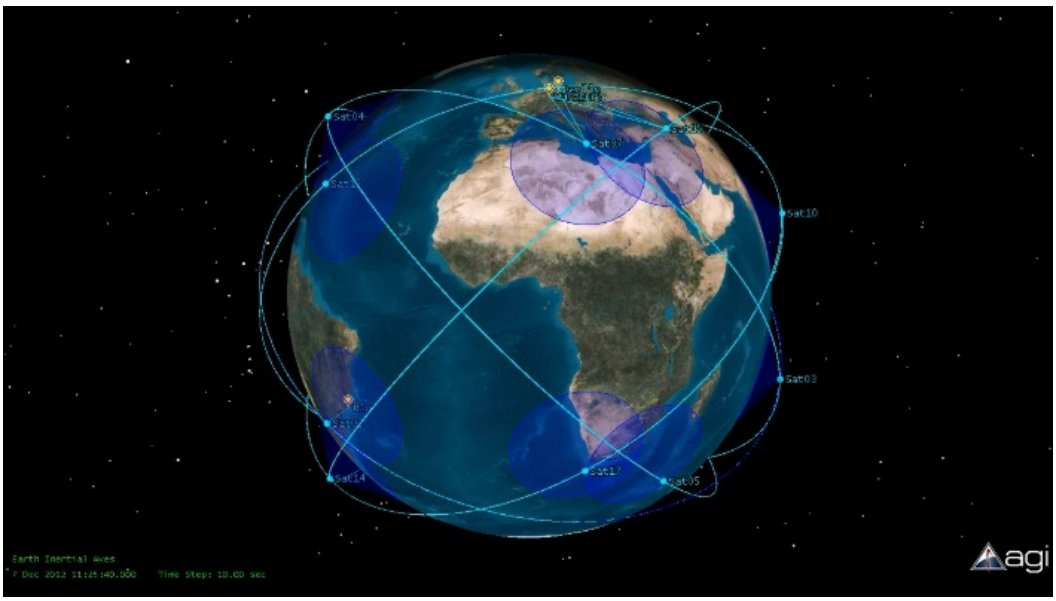


Fig. 4.0.: 3D model of a $45° : 18/6/0$ Walker constellation for traffic monitoring.

4.3 Evaluation

4.3.0 Model

For the evaluation of the approach described above, a global ship distribution as collected in the PASTA MARE project [132] was considered. In order to achieve a high temporal resolution for traffic monitoring, a $45° : 18/6/0$ Walker constellation was designed (see Fig. 4.0). This constellation allows for a high temporal surface coverage except for polar regions, where ship distribution is naturally rather sparse. The second constellation used for the simulation is a $66° : 18/6/0$ Walker constellation. An optimal footprint diameter has been determined by considering the total number of messages which can be received during the simulation interval. As a result an antenna opening angle of $120°$ was used for this constellation. Therefore, it is assumed that signals can only be received within this opening angle of the nadir pointing satellite antennas. This assumption requires the availability of attitude control systems on board of each satellite. The applicability of such systems was demonstrated, even for one unit CubeSats, as described in [57]. For the inter-satellite communication antennas an omnidirectional characteristic is assumed, which can be achieved either by using dipoles or by using multiple antennas. The selection of the orbit altitude of the constellation is a trade-off between the received signal power and the atmospheric drag. A low altitude leads to short mission lifetimes or high propellant requirements due to the atmospheric drag. High altitudes lead to high communication distances and therefore higher bit error rates. As a compromise an orbit altitude of $700\,\mathrm{km}$

was selected. For evaluation the monitoring system is complemented by up to four globally spread ground stations, used for collecting monitoring data from the satellites and making the data available for end users. The performance of the observation system was evaluated by analyzing the contact windows of the ship-to-satellite links, the inter-satellite links, and the satellite-to-ground links. Since maximum communication distances are highly dependent on frequency, antenna gain, available transmission power, and many other parameters, the impact of different radio ranges is considered as well.

AIS message reception

Using a $1° \times 1°$ tessellation of the earth's surface, we work with a fixed distribution of $N_i = \lceil N_i^{\text{PastaMare}} \rceil$ ships in the i^{th} tile, yielding a total of approximately 68,000 ships. For simplicity, we assume each ship to stay within its tile during the $24\,\text{h}$ time interval that was simulated. We further assume that each ship transmits position reports once every $10\,\text{s}$ on average, which is the reporting interval for AIS Class A ships heading at speed up to $14\,\text{kn}$ while not changing course. Satellites receive primary AIS messages via an omnidirectional antenna with a sharply defined sensor opening angle of $120°$ full cone. We use a simplified model where each AIS message that is transmitted by a ship within the satellite's footprint is successfully received if and only if no other AIS message is sent from a different ship within the sensor footprint in the same SO-TDMA slot and channel. We thereby neglect the limited signal power in the vicinity of the transmitting ship's zenith as well as the noise caused by AIS transmissions from ships visible to the satellite but outside the sensor footprint.

Inter-satellite links

Let t_i^s be the starting time of the s^{th} frame sent by the i^{th} satellite. This frame is successfully received by the j^{th} satellite if and only if $\text{dist}_{ij} < d_{\max} \wedge \forall k \forall r : (\text{dist}_{jk} > d_{\max} \vee |t_i^s - t_k^r| > T)$, hereby treating collisions explicitly. Because of the pure ALOHA medium access scheme used, the negligence of signal propagation delay is not assumed to cause any systematic bias in transmission probability. However, we assume to slightly overestimate the achievable throughput for inter-satellite communication by neglecting the noise level caused by transmissions of satellites further apart than $d_{\max}$.

Simulation

The simulation incorporates the reception of primary AIS messages, forwarding of AIS data records between satellites, as well as forwarding AIS data to one or more ground

stations. A period of 24 h is simulated where all satellite databases are initially empty; therefore the first 6 h of the simulation are used for warmup only and data is measured during the remaining 18 h.

The simulated time domain is homogeneously discretized in steps of 60 s each. A single time step is simulated as follows:

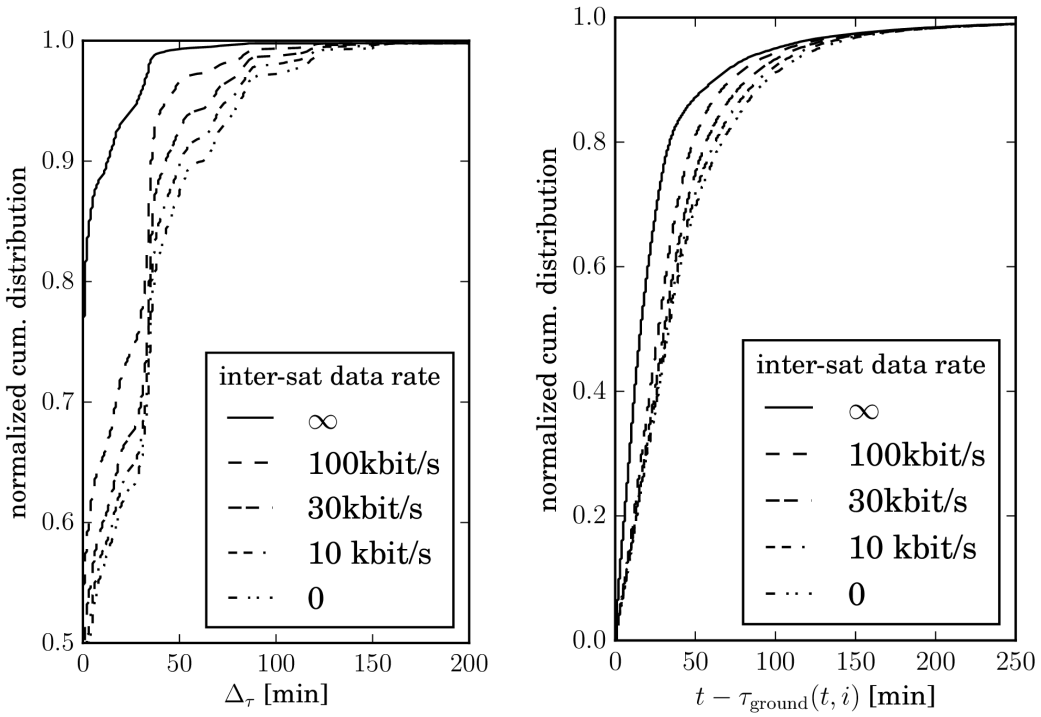
- Assuming an unlimited downlink bandwidth, each ground station's database is updated using the databases of all satellites within its field of view immediately during GS contact periods.
- For each satellite i the set of ships s_i^{fp} within the sensor footprint is determined. For every ship in s_i^{fp} the probability that at least one AIS message is successfully received by the satellite within the current time step is, following the results from [130], given by (4.0),

$$P = 1 - \left[1 - \left(1 - \frac{N_i^{\text{vis}}}{75\,\text{Hz} \cdot M \cdot \Delta T}\right)^{M-1}\right]^{\frac{T_{\text{obs}}}{\Delta T}} \tag{4.0}$$

  where $T_{\text{obs}} = 60\,\text{s}$, $\Delta T = 10\,\text{s}$ is the assumed reporting interval, $N_i^{\text{vis}} = |s_i^{\text{fp}}|$ is the number of visible ships, and M is the number of organized SO-TDMA cells formed by these ships.
- N_i^{vis} independent Bernoulli distributed pseudorandom numbers determine the subset $s_i^{\text{vis}} \subseteq s_i^{\text{fp}}$ that is actually detected by the i^{th} satellite in the current time step. This procedure is performed for every satellite independently.
- For each satellite the inter-frame times within the corresponding time step are determined by exponentially distributed independent pseudorandom numbers. Chronologically for each transmission a random subset of the AIS records in the sender's database is selected and used to update the databases of all neighbor satellites unless inhibited by a packet collision.

4.3.1 Results

Let $\tau_{\text{orbit}}(t, i)$ be the newest time stamp of all AIS messages from vessel i received by any satellite at or prior to time t and let $\tau_{\text{ground}}(t, i)$ be the corresponding newest time stamp of AIS messages from vessel i that is forwarded to any ground station prior to time t. Then, $\Delta_\tau(t, i) := \tau_{\text{orbit}}(t, i) - \tau_{\text{ground}})$ is the delay caused by non-perfect network connectivity (inter-satellite and/or between satellites and ground, in the following called delivery delay), whereas $t - \tau_{\text{ground}}(t, i)$ is the total monitoring delay that is also caused



(a) Delivery delay.

(b) Monitoring delay.

Fig. 4.1.: Relative cumulative counts of delays.

by imperfect ground coverage and AIS message reception probability. Our aim here is to reduce the delivery delay (and thereby the monitoring delay) through inter-satellite message forwarding. To demonstrate the applicability of our protocol, we collected values for $\Delta\tau$ for all vessels over the measurement time interval of 18 h by simulation.

In the evaluation, we assume that satellites deliver their database to Earth via a separate communication channel whenever they come into a ground station's field of view. Even though this assumption is a strong simplification of the real situation, it still allows for direct comparison to a system without inter-satellite communication, which is equivalent to the case where the inter-satellite data rate is zero. We compare the cumulative delivery delay distribution for different inter-satellite data rates in Fig. 4.1a. The curves corresponding to finite bit rates lie well in the middle between the case without inter-satellite links and the idealized case of an infinite inter-satellite data rate, where any two satellites exchange all information instantaneously whenever they are within communication range. Clearly, the use of inter-satellite links significantly reduces the typical delivery delays even at relatively low data rates. Fig. 4.1b shows the distribution of the total age of the newest record received on ground $t - \tau_{\text{ground}}(t, i)$, i. e., the full monitoring delay.

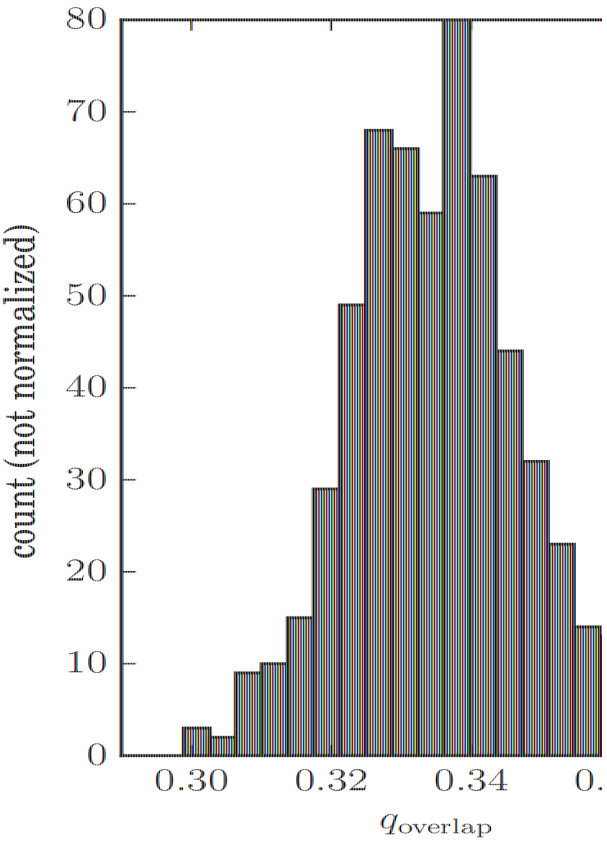


Fig. 4.2.: Histogram of database overlaps when two satellites meet.

When two satellites come within range to communicate, their databases will typically partially overlap. Therefore, the transmission of randomized subsets of the satellites' databases causes an overhead due to transmitting data that is already known at the receiver.

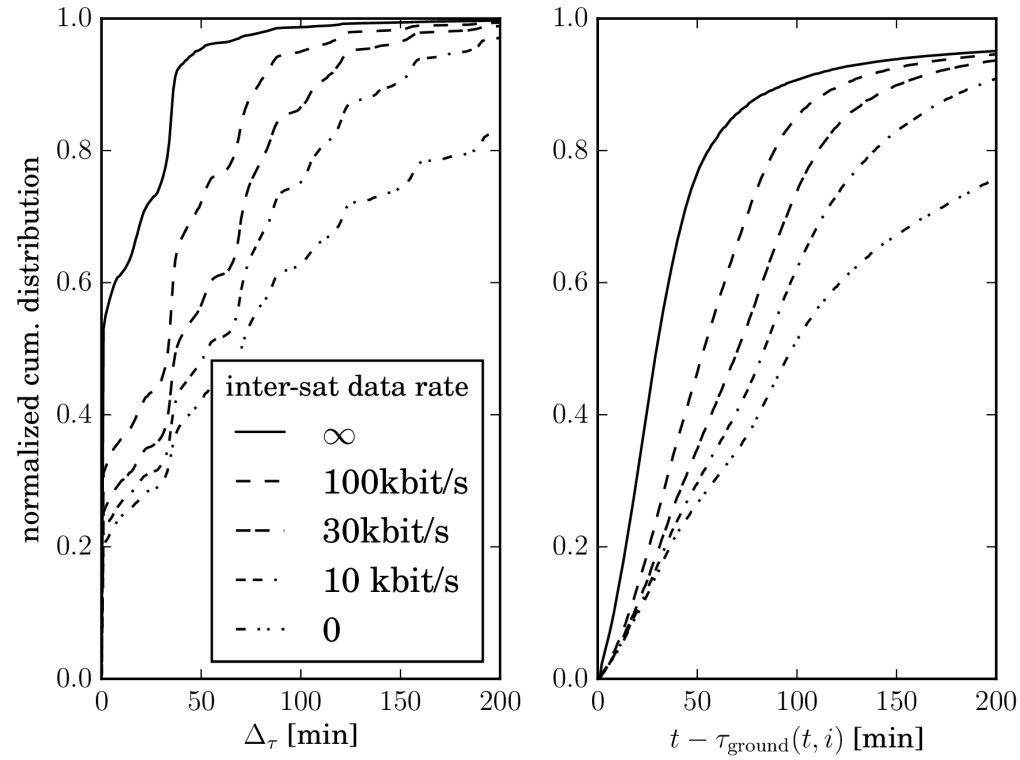
To estimate this overhead we histogrammed (Fig. 4.2) the distribution of $q_{\text{overlap}}(i, j)$, which is the fractional overlap between the databases of two satellites. The data is collected from all satellite meetings within an $18\,\text{h}$ period, in a simulation with $30\,\text{kbit}\,\text{s}^{-1}$ transmission rate. The result given in Fig. 4.2 suggests a typical overlap of $30\,\%$ to $40\,\%$, even though with different parameter settings the overlap was observed to range up to $60\,\%$. Given this overlap, one might argue that message forwarding could in some situations benefit from additional coordination, but not by much when considering the overhead of a then required handshake. As expected, the benefit of data forwarding increases as the number of ground stations decreases.

In Fig. 4.3 the distributions of delivery delay and total monitoring delay in the case of one single ground station is shown for both the original $45°$ constellation and the alternative constellation with $66°$ inclination. To investigate the effects of the somewhat optimistic assumption that any two satellites within visibility range can communicate, which corresponds to $d_{\text{max}} \approx 6200\,\text{km}$, the communication range was varied for a fixed communication bandwidth of $30\,\text{kbit}\,\text{s}^{-1}$ and the resulting delivery delay distribution is shown in Fig. 4.4. We note that for $d_{\text{max}} \geq 3000\,\text{km}$ the data forwarding performance is not dramatically reduced.

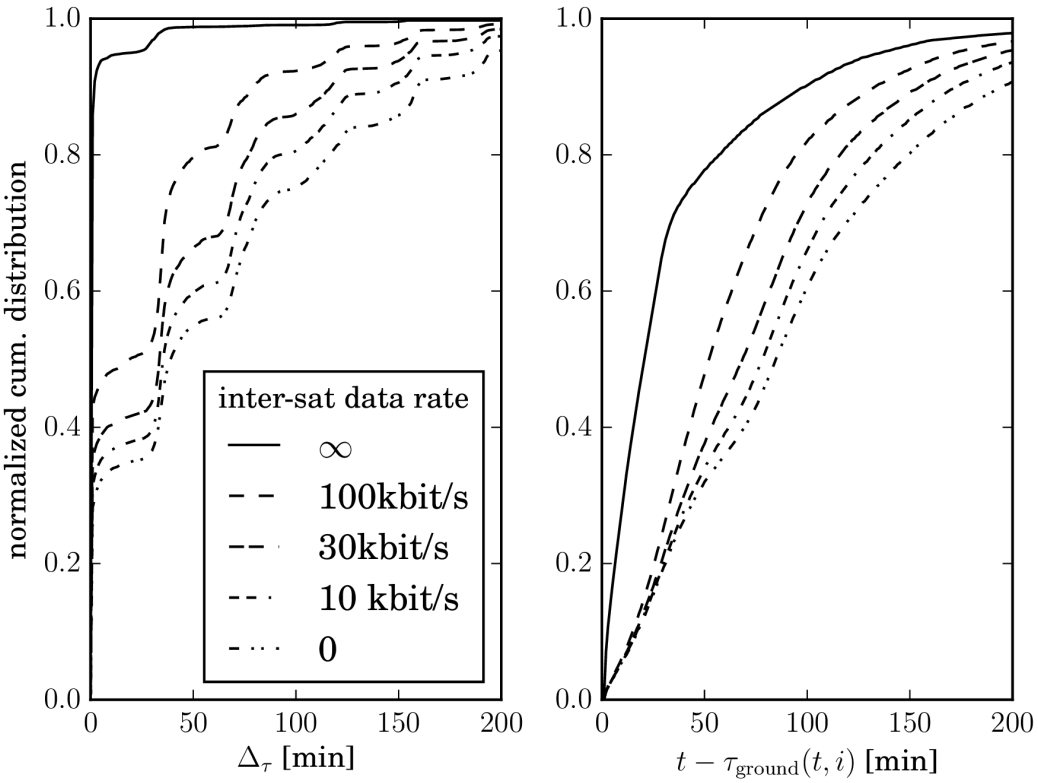
Finally, we demonstrate the robustness of the system with respect to loss of individual satellites by omitting a subset of satellites during the whole simulation. For data rates of 0, $30\,\text{kbit}\,\text{s}^{-1}$, and ∞ we simulated the loss of one and two satellites each with several different subsets of omitted satellites. The resulting (monitoring- and delivery-) delay distributions (Fig. 4.5) show that the relative location of the distribution corresponding to a data rate of $30\,\text{kbit}\,\text{s}^{-1}$ between the limiting cases does not change significantly. We also note that the apparent benefit of delivery delay from defunct satellites for small delay times is not an error but is instead a consequence of the definition of that observable.

4.4 Conclusion

In this chapter we briefly demonstrated how satellite-based real-time traffic monitoring of terrestrial vehicles can benefit from inter-satellite communication in terms of a significantly reduced delays. Because of its stateless and stochastic design the system is expected to be inherently robust, flexible and scalable with respect to failing and newly added satellites,



(a) 45° inclination.



(b) 66° inclination.

Fig. 4.3.: Delivery- and monitoring delay in case of a single ground station.

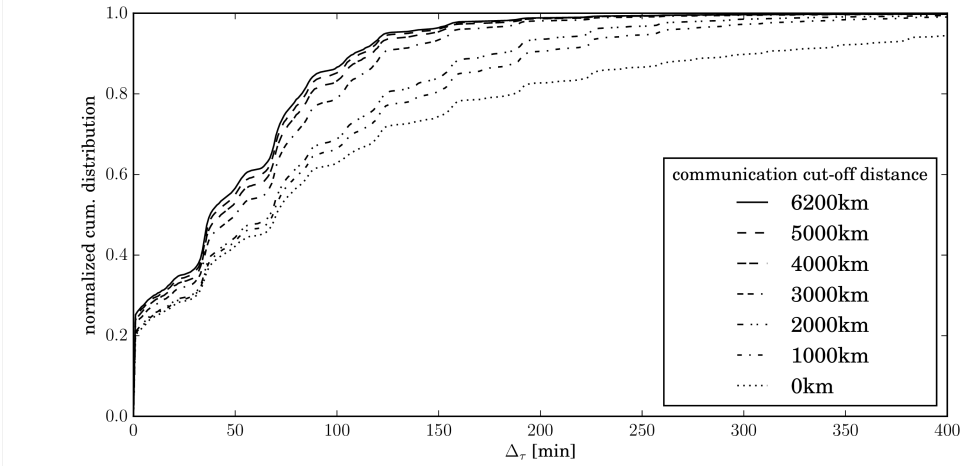


Fig. 4.4.: Delivery delay cumulative distribution for different communication ranges and a bandwidth of 30 kbit s^{-1}.

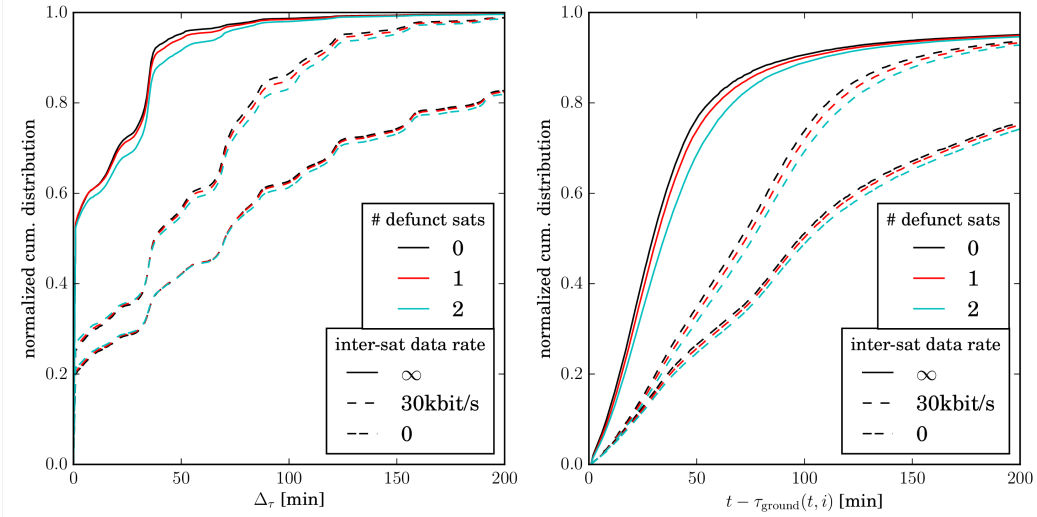


Fig. 4.5.: Delivery (left) and monitoring (right) delay distribution for different numbers of defunct satellites.

and also with respect to unreliable communication channels. The presented approach is an enabling technology for a variety of traffic monitoring applications.

5 More Efficient LEO-Satellite Downlinks Using Distributed Source Coding

> *Things are only impossible until they're not!*
>
> — **Captain Jean-Luc Picard**
> from: "When the Bough Breaks" (15 February 1988) written by Hannah Louise Shearer

This chapter is largely based on our own publication *"Efficient Multi-Satellite Downlinks for Earth Observation Data Based on Distributed Arithmetic Coding"*[1] published as a full paper at LCN'19.

TL;DR *Distributed Arithmetic Coding is a viable choice to exploit inter-dataset redundancies that occur when multiple satellites in similar Earth orbits measure the same kind of data.*

5.0 Introduction

Earth observation is a prevalent goal of low-Earth-orbit satellite missions, including missions based on nano satellites like CubeSats [133]. Due to their low cost, nano satellites lend themselves to being launched and operated in formations of identical satellites within one mission, which increases fault tolerance, flexibility, reconfigurability, and upgradability. The small form factor, however, leads to limited energy production capabilities and prohibits the use of large, high-gain antennas. This, together with short and infrequent ground station contact periods typical for low Earth orbit, makes the downlink to the GSs a primary bottleneck for the acquisition of mission data.

When two (or more) satellites of a mission measure the same kind of data in orbit it is not uncommon that the data acquisition by different satellites is correlated. By making use of this (information-theoretic) inter-dataset redundancy, the satellites' downlink efficiency can be improved. As theorized by Slepian and Wolf [58], such redundancy of correlated datasets located at two or more independent sources can be exploited to reduce the amount of data that needs to be transmitted, without a need for the sources to communicate with

each other. That is, there is no need to exchange information between the satellites. Yet the total amount of channel bandwidth needed for transmission down to Earth could be reduced compared to compressing, i. e., encoding and decoding, each transmitted dataset independently. The challenge to code different sources' data accordingly is usually referred to as the distributed source coding (DSC) problem.

In this chapter, we investigate the challenges associated with efficient utilization of multi-satellite downlink capacity in such a setting. Though many solutions to the DSC problem have been proposed, they have been designed and studied mostly in terms of pseudo-random data of known entropy and correlation. In order to harness these theoretical findings for practical system designs, a couple of additional challenges need to be overcome. The satellites' restricted energy budget and limited processing power requires a simple, lightweight encoder. The lack of an exact statistical model of the encoded real-world data requires additional flexibility of the ground station's joint decoder. Building upon DAC [59] as a prototypical DSC approach, we discuss how these requirements can be addressed. We describe a protocol with an adaptive joint decoder, which scales its decoder state as needed. This decoder is operated with different fitness functions, thereby paying tribute to the uncertainty in the correlation of data acquired by real-world sensors.

Using Earth magnetic field data obtained during the MagSat [15] mission, we evaluate our proposed architecture. We show that inter-dataset redundancy can indeed be utilized to improve the coding rate for Earth observation satellite downlinks. To the best of our knowledge, we are the first to successfully apply DSC techniques to a complex, distributed real-world setting with an extensive performance evaluation based on more than 400 pairs of correlated datasets.

The remainder of this chapter is structured as follows: after Section 5.1 on related work, we discuss the opportunities and requirements that are characteristic of efficient down-link utilization in multi-nano-satellite missions in Section 5.2. In Section 5.3 we review the principles of (distributed) arithmetic coding. A codec that is tailored to the specific needs of the use case considered is developed in Section 5.4, followed by Section 5.5 in which we present an evaluation of our codec. Finally, we draw a conclusion in Section 5.6.

5.1 Related Work

An important challenge when working with nano-satellites is their limited energy supply and low computing power. This problem is also encountered in Earth-bound wireless sensor networks (WSN). Studies of WSN nodes have shown that the power used for data transmission amounts to about 80 % of the total energy consumption [60]. As the power

consumed for data transmission is proportional to the amount of data transmitted, a common strategy is to reduce the size of the data by compression [61].

In [62] it is pointed out that compression algorithms potentially consume more energy during their execution than the reduced amount of transmitted data can save later on. Orbit-to-Earth communication is, however, very energy consuming. Therefore, we argue that compression is worthwhile in this context. Still, we take into account the argument made in [62] and avoid overly complex compression algorithms.

The theoretical basis for distributed source coding was laid by Slepian and Wolf [58] in 1973. They proved that it is theoretically possible to losslessly compress two correlated datasets encoded separately by sources not communicating to each other to an overall size that is smaller than what could have been achieved if each dataset was compressed independently. In order to achieve this, the encoders as well as the joint decoder only need a priori knowledge about the datasets' correlation.

Today, a wide range of algorithms aiming to make practical use of Slepian and Wolf's findings is known [63, 64, 65]. However, many of these algorithms are based on channel codes, e. g., [63, 65], which have some severe drawbacks. The following problems are found [66] to be inherent to DSC based on channel codes: they are unable to reach optimal performance for short datasets. The symbols in a dataset are expected to follow a stationary distribution, which is often not the case in practice. Furthermore, it is argued that many of the algorithms are complex and may even increase the encoder's overall energy demand [66].

To overcome these shortcomings, in 2007 the first DAC algorithm was proposed [66], an algorithm based on arithmetic coding [67] and quasi-arithmetic coding [134], for the primary and secondary sources respectively. This gave rise to new approaches to distributed source coding based on arithmetic coding [68, 69, 70], new models to describe the decoding complexity of DAC [71], and approaches to avoid decoding errors, for instance by including forbidden symbols [72] or a special end-of-file symbol [73]. In DAC literature, often only binary, memoryless data sources are considered, e. g., [74, 69, 68]. The use of more complex sources, like for example Gaussian distributions [75] or a Markov-1 source [76, 70], is very rare. Even then, only memoryless sources are used in [75] and the Markov sources in [76] and [70] emit only binary symbols. While channel-coding-based distributed source coding has already been evaluated for distributed real-world settings, see e. g., [77], DAC algorithms have not.

To the best of our knowledge, only in [78] DAC was used to compress real-world data so far. In contrast to our work, however, a traditional compression problem is considered there, in which a single dataset is compressed and DAC is solely used to leverage correlations within

this dataset. The same holds for [79] where scalar coset codes, a low complexity DSC technique, to compress hyperspectral images losslessly in a non-distributed setting, are used. Because we use DAC to compress correlated datasets in a distributed setting, where each encoder has access to its own dataset only, we face radically different challenges.

A general problem shared among all DSC techniques is that so far no approach can guarantee lossless compression. However, if decoding errors can be detected, they may still be overcome by DAC when additional resources, e. g., main memory, can be provided.

What further sets DAC algorithms apart from both classical lossless compression algorithms – like arithmetic coding [67] or Huffman codes [80] – *and* classical lossy algorithms, is that a major part of the decoder can be exchanged without needing to change the encoder as well.

5.2 Efficient Coding of Correlated Data

We consider a scenario of two nano satellites in low Earth orbit, each acquiring measurement data using some instrument. The satellites cannot communicate with each other, but both transmit the acquired data to a common satellite ground station. The satellites' possible measurement outcomes (i. e., the data) can be seen as a pair for random variables X and Y whose probability distributions are unknown. To use communication links efficiently, one usually resorts to source coding, also known as lossless data compression. A dataset $x \in X$ that is to be transmitted, e. g., from a satellite to a ground station, can often be encoded using code words that are on average shorter than the corresponding plain representation, thereby utilizing less channel capacity. The expected code word length is, according to Shannon's source coding theorem [81], lower bounded by the data sources' *entropy* $H(X)$, which in turn depends on the probability distribution of X. For simple scenarios, where a source is known in advance to be a sequence of stochastically independent random variables or a Markov chain, the entropy may be computed analytically. So-called entropy codes are known that can be used to encode this data at an expected code word length, or *rate*, R_X close to the entropy. When coding physical measurement data efficiently, neither the sources' probability distribution nor its entropy are known, and therefore heuristic codes are used in practice. The PNG image format [135] for example uses a codec based on the simple assumption that neighboring pixels have similar values; using a combination of differential coding and entropy coding, typical photographic image data can be losslessly transmitted using code words that are significantly shorter than the raw pixel data. Optimization of code word length is, however, not the only requirement of source coding in the context of effective utilization of channel capacity: especially in scenarios where the source is limited in terms of computational and electrical power, like

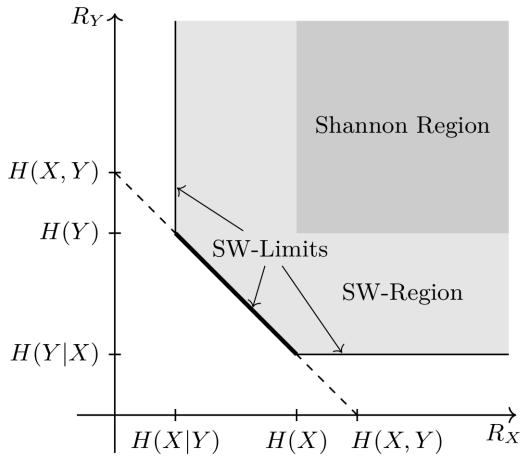


Fig. 5.0.: The Slepian-Wolf-Region and the Slepian-Wolf-Limits for two data sources.

nano satellites, it is reasonable to use a simple, fast encoder that achieves only suboptimal coding rate in favor of low complexity.

We consider a scenario of two data sources X and Y that are correlated, i. e., X and Y are not stochastically independent. As pointed out by Slepian and Wolf [58], the additional inter-dataset redundancy may be exploited to reduce the coding rates further than what can be achieved by independently source-coding and decoding X and Y. The set of feasible rate combinations shown in (5.0), the Slepian-Wolf region, is a convex subset of the Cartesian plane shown in Figure 5.0.

$$R_X \geq H(X|Y), \qquad R_Y \geq H(Y|X), \qquad R_X + R_Y \geq H(X,Y) \tag{5.0}$$

$H(X,Y)$ and $H(X|Y)$ denote the joint and conditional entropy respectively. Just as entropy codecs can be used to reduce a single source's rate close to its entropy, DSC codecs have been proposed to reduce the sources' combined code rate near to the Slepian-Wolf bound (see Section 5.1).

Our goal is to economically apply this coding-theoretical knowledge on real-world measurement data. Using a simple encoder based on DAC, coding rates can be reduced compared to classical source coding while additional computational complexity is added only at the decoder side where resources are not as limited.

5.3 DSC Based on Arithmetic Coding

Our proposed approach is based on arithmetic coding (AC) and DAC [59]. One satellite (Y) conventionally compresses its dataset y losslessly such that the ground station can

decode it without a need for further information. The other satellite (X) applies DAC on its dataset x, to reduce the size of the corresponding code word $\hat{x}$ even more before its transmission. The joint decoder, i. e., the ground station, cannot decode $\hat{x}$ on its own. It can, however, decode $\hat{x}$ and thus fully reconstruct x when using the decoded correlated dataset y together with some knowledge about the correlation of x and y. We note that even though asymmetric in terms of the achievable code word lengths, this approach can still be used to symmetrically increase coding efficiency of both satellites' downlinks by alternating the sources' roles. In the following, we briefly recall the key concepts of AC and DAC and highlight why DAC is particularly well suited to the use case considered here.

5.3.0 Arithmetic Coding

AC [67] is an entropy coding technique that encodes a sequence of symbols based on an internal statistical model of symbol probabilities. If each symbol's probability is known and depends only on the symbols prior to that, a rate close to the entropy can be achieved in theory [136].

AC does not represent the input symbols $x = (x_1, \dots, x_k)$ by a sequence of individual code words, but instead represents the whole sequence as *one* code word. By repeatedly subdividing and shrinking $[0, 1)$, the encoder maps the input sequence to a *characteristic interval* $C \subseteq [0, 1)$. Each encoded symbol shrinks the interval by a factor equal to its probability. For details see, e. g., [67].

The characteristic intervals correspond to the possible sequences of length k and form a partition of $[0, 1)$. Given k, any real number in C is characteristic for x. Thus, any $w \in C$ can be chosen as the *code point* of x, and its binary representation is the compressed code word $\hat{x}$. The length of $\hat{x}$ can generally be assumed to be about $-\log_2|C|$ bits [59], where $|C|$ denotes the length of C.

To retrieve x from $\hat{x}$, the decoder simply replays the encoding procedure by iteratively determining and emitting the unique symbol that results in a shrunk interval still containing w until all k symbols are emitted.

Even though this process appears to require floating point arithmetic with infinite precision, different strategies have been found to circumvent this problem. AC can be implemented with an encoder using only integer arithmetic and a small constant number of integers as state [82]. This makes AC a perfect candidate for encoding in embedded systems like nano-satellites.

As stated above, AC gives good compression results if a good statistical model is available. Models used in the literature range from simple symbol frequency counts and Markov chains to models where the last n symbols are taken into account [83, 84]. Because accessing memory is energy consuming [60], overly large models should be avoided in our use case.

5.3.1 Distributed Arithmetic Coding

The encoding of DAC [59] is nearly identical to the encoding of AC. The subdivide-and-shrink procedure is identical, only the subintervals corresponding to different symbols are enlarged, so that they overlap with their neighbor intervals. The enlarged subintervals in turn lead to a larger characteristic interval C and therefore to shorter code words. They also lead to a very specific form of ambiguity, because a code point does in general no longer belong to only one valid code word. In order to resolve the arising ambiguities, we use side information from the other satellite and domain knowledge about correlations.

In line with the arguments above, we use a simple strategy where each symbol's subinterval is increased by the same constant *overlap factor* $c > 1$. The product of the symbol's probability and c, called *extended probability* in this work, determines the size of the symbol intervals. We used the following simple strategy to enlarge every symbol's interval $[p_{\text{low}}(s), p_{\text{high}}(s))$ by a factor of c, giving the new interval $[l(s), h(s))$:

$$\begin{aligned} h(s) &= \min\left(p_{\text{low}}(s) + c \cdot p(s),\ 1\right), \\ l(s) &= \begin{cases} p_{\text{low}}(s) & \text{, if } h(s) < 1 \\ \max\left(p_{\text{low}}(s) + 1 - c \cdot p(s),\ 0\right) & \text{, otherwise.} \end{cases} \end{aligned} \tag{5.1}$$

This reduction of code-word length by approximately $k \log_2 c$ bits comes at the cost of a small constant number of arithmetic instructions per encoded symbol, thereby increasing the computational cost of encoding by at most a small constant factor, depending on the implementation. We assume this additional cost to be affordable. When using look-up table-based implementations of AC, as in [134], c would affect only table creation, thereby eliminating any per-symbol computation overhead in the encoder.

On the decoder side, the challenge is to resolve the deliberate ambiguities. Whenever the decoder encounters the code point in a region where two symbol intervals overlap in one or more decoding steps, the correct continuation of the decoding process is ambiguous. It is up to the decoder to jointly use the code point w_x as given by $\hat{x}$, the decoded side information y, and a model of the correlation of X and Y in order to reconstruct x. Thus, the decoding process, which is described in the next section, is of substantially increased

complexity compared with conventional arithmetic coding. This is the price for the additional compression of x beyond the entropy limit.

Note that the resulting asymmetry in coding complexity—nearly unchanged encoding effort, but increased decoding effort—matches the asymmetry of resource availability as found in satellite missions very well: on-board resources are scarce, while additional computation on the ground after transmission is easy.

5.4 A Flexible Distributed-Arithmetic-Coding-Based Codec for Correlated Time Series Data

Each time the decoder encounters an ambiguity, there are at least two different possibilities to continue decoding. The decoder needs to use the known side information y to decide which decoding path is more likely. One can assume that in most practical applications only limited knowledge about the correlation of the data sources is available. Because *every* ambiguity encountered must be resolved faultlessly to decode the dataset correctly, a decoder could try to pursue *every* possible continuation. This, however, is not always feasible as the number of possible solutions grows rapidly with the length of the encoded sequence. Instead, *pruning* is used to limit the number of pursued solution paths.

A practical solution to handle the rapidly growing number of decoding paths is pruning with the so-called M-Algorithm [85]: a fixed upper bound for the number of paths pursued by the decoder is defined and paths are removed by the decoder whenever their number exceeds this limit. To decide which paths to keep and which to prune, a *fitness function* is used to evaluate how well each partial decoding result fits the assumed correlation with the side information y.

The fitness functions proposed in literature are one main reason for the difficulty of applying DAC in practice. Usually, these fitness functions are based on known joint or conditional probability distributions of individual pairs of symbols, e. g., in [68, 66]. In particular, the so-called *Maximum A Posteriori* [59] (MAP) metric is frequently used [72, 86, 75]. In this work, MAP is not employed, because we deem the necessary detail of knowledge about the datasets' distribution unrealistic. Instead, we use fitness functions focusing on more general properties of the correlated datasets.

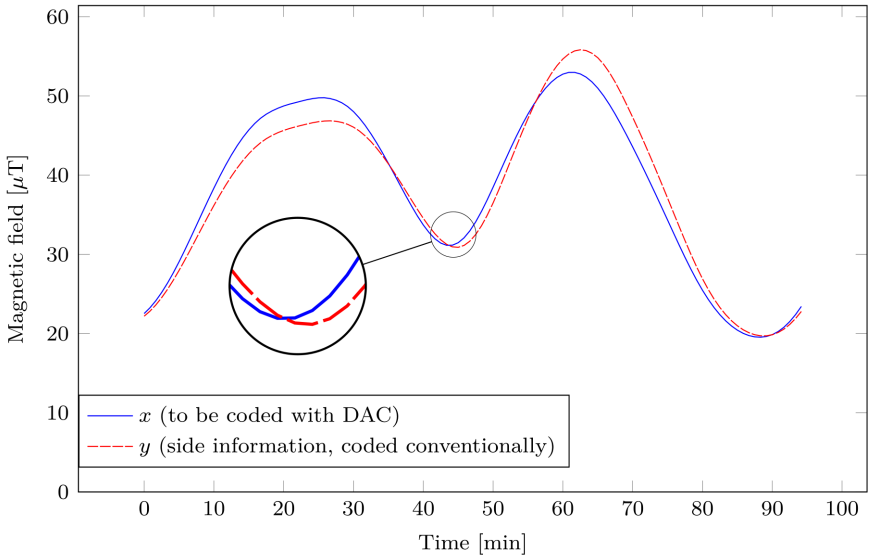


Fig. 5.1.: Example for two correlated datasets of Earth's magnetic field data.

5.4.0 Lightweight General-Purpose Encoding of Time-Series Measurement Data

In our use case scenario we apply DAC to the encoding of Earth's magnetic field data. The datasets are correlated pairs of scalar time series sampled at a fixed rate. A pair of example datasets is shown in Fig. 5.1.

To keep the computational complexity manageable on a small satellite platform and to broaden the applicability of our approach, we abstained from trying to create the best possible compression algorithm for the specific datasets considered. Instead, we use a two-step encoder consisting only of differential coding for de-correlation and arithmetic coding. A quantized sampled scalar time-series of length k is a sequence x of integers. This is first transformed by taking successive differences:

$$x_\Delta = (x_1, x_2 - x_1, x_3 - x_2, \dots, x_k - x_{k-1})$$

The symbols in x_Δ are then, according to their frequencies, encoded using AC in case of Y or DAC in case of X, using a fixed overlap factor c. This encoding is lightweight, general purpose, yet tailored to encode time-series data: the magnetic field strength measured by a satellite while traveling in its orbit is a continuous real-valued function of time. Being sampled at a fixed rate, it is a plausible assumption that consecutive samples are highly correlated. No further assumptions are made, reducing the risk of overfitting our model to the specific datasets. The finite precision AC (or DAC) encoder we used requires only a

small constant amount of memory and its run-time is linear in the number of encoded symbols [134].

5.4.1 Adaptive Decoding

We propose a new kind of decoder, called *adaptive* decoder, which extends the classical breadth-first M-Algorithm [85]. The latter uses a fixed *pruning threshold* M to determine when and how many paths to prune. Pruning starts as soon as the number of paths surpasses $2 \cdot M$ and then M paths are pruned at once.

In the literature M is chosen arbitrarily, e. g., $M = 2048$ in [59]. However, if M is too small, the correct path may be pruned during decoding and thus the dataset cannot be decompressed. If M is too large, i. e., a smaller M would have sufficed, the decoding process wastes resources. We therefore avoid a fixed value for M and instead adjust the threshold dynamically during decoding.

To this end, we include a hash value $h(x)$ in the header of the transmission of $\hat{x}$. $h(x)$ can be used to confirm if a given guess is correct (with a properly chosen hash function the probability of hash collisions is negligible). Our adaptive decoder starts with a small pruning threshold (e. g.$M = 1$), and increases it iteratively on demand. For each value of M, the M-Algorithm is executed, yielding between M and $2M$ candidates of k symbols each. If one of these candidates can be confirmed to be the correct dataset x using the hash function, the decoding terminates. Otherwise, M is doubled and a new decoding attempt is made. This way, decoding time is saved *and* resource limitations are taken into account. The iteration is, in addition, terminated by an upper bound M_{max} that depends on computing resources available as well as on the length of the hash function h.

The computational cost of iteratively re-executing the M-Algorithm until M_{max} is reached is at most twice as high as the worst-case cost of the final iteration with $M = M_{\text{max}}$ as can be seen by the following argument: since the worst-case complexity of a single run of the M-Algorithm is at least linear in M, the M-Algorithm's worst-case runtime will at least double with each iteration. Therefore, the final iteration costs at least as much as all prior iterations together (in terms of worst-case complexity).

5.4.2 The Fitness Function

We now turn to the description of the methods we employed to exploit the similarities between the side information and the correlated dataset in the design of our fitness functions.

It is difficult to state a universal strategy of how the knowledge about inter-dataset correlation is obtained at the decoder site, as the nature and extent of such knowledge depends on the satellite mission, the type of data measured, and a priori knowledge about the phenomena being observed. If, at the beginning of data collection, the decoder has no knowledge at all, one could begin the mission with an exploration period where data is transmitted without inter-dataset redundancy removal, so that correlation knowledge can be built from that.

In general, a fitness function has to be designed with the similarities of the correlated datasets (or data sources) in mind. At its core, a fitness function evaluates a property that is typical for the underlying datasets but is less likely to be found in data generated by incorrect decoding. For the Earth magnetic field data used here, we used fitness functions that either compare the smoothness of the datasets or analyze the similarity of the curve shapes. In addition, one could, in principle, extend the adaptive decoder to use multiple different fitness functions within one decoding run to further increase the decoding probability. This, however, would introduce many additional degrees of freedom and is left to be studied in future work.

In our algorithm, each decoding path's fitness value is updated every time it is extended by a new symbol s. Because wrong paths may share a potentially long prefix with the correct decoding path, we use a windowed approach for all fitness functions and only compare the newest W symbols (including s) of the decoding path with the corresponding W symbols in the side information. The cumulative sum of these quantitative comparisons is then used as the decoded path's fitness value.

Let $g(x, y, W, k')$ be the function evaluating the similarity of x and y for a window size W after the symbol $x_{k'}$ was decoded. We define x's fitness value to be

$$f(x, y) = \sum_{k'=1}^{k} g\left(x, y, W, k'\right).$$

Therefore, it is the similarity function g that sets the different fitness functions apart.

First, we describe the similarity function of a fitness function comparing the smoothness of x and y ("FitPCC"). Let g_{FitPCC} be this similarity function:

$$g_{\text{FitPCC}}(x, y, W, k') = -\left(\left|\rho_{\text{PCC}}(x, W, k')\right| - \left|\rho_{\text{PCC}}(y, W, k')\right|\right)^2,$$

where $\rho_{\text{PCC}}(x, W, k')$ is the autocorrelation of x or y respectively, in a window of size W ending at the k'th symbol expressed via the Pearson correlation coefficient. Let $\hat{\tau} = \max(1, k' - W)$, then

$$\rho_{\text{PCC}}(x, W, k') = \frac{\sum_{i=\hat{\tau}}^{k'-1} (x_i - \bar{x}) \cdot (x_{i+1} - \bar{\dot{x}})}{\sqrt{\sum_{i=\hat{\tau}}^{k'-1} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=\hat{\tau}}^{k'-1} (x_{i+1} - \bar{\dot{x}})^2}},$$

where $\bar{x} = \frac{1}{k'-1-\hat{\tau}} \sum_{i=\hat{\tau}}^{k'-1} x_i$ and $\bar{\dot{x}} = \frac{1}{k'-1-\hat{\tau}} \sum_{i=\hat{\tau}}^{k'-1} x_{i+1}$.

Next, we describe the similarity function g of the fitness functions FitSlope and FitPresent, which evaluate similarities between the curve shapes: Let g_{FitSlope} be the similarity function of FitSlope and $\tau = \max(1, k' - W + 1)$, defined as follows:

$$g_{\text{FitSlope}}(x, y, W, k') = -\left((x_{k'} - x_\tau) - (y_{k'} - y_\tau)\right)^2. \tag{5.2}$$

And let $g_{\text{FitPresent}}$ be the similarity function of FitPresent and $\tau = \max(1, k' - W + 1)$, given by

$$g_{\text{FitPresent}}(x, y, W, k') = -\sum_{i=\tau}^{k'-1} (x_i + d_c - y_i)^2, \tag{5.3}$$

where $d_c = y_k - x_k$ is the offset between x's most recent symbol and the corresponding symbol in the side information y.

For all aforementioned fitness functions, the window size W strongly influences how easily the correct and incorrect decoding paths can be discerned. The choice of W will be discussed in the next section.

5.5 Results

Before we evaluate the *adaptive* decoder, we briefly discuss the statistical model used for evaluation.

5.5.0 Implementation

Because our focus is on evaluating DAC's applicability to real-world problems, we do not aim for perfect compression of the magnetic field data in the sense of exactly reaching the

Slepian-Wolf limit.[4] We make use of the autocorrelation of the datasets by transmitting only the differences between consecutive magnetic field values. In coherence with the findings of [62], we use a simple frequency-based zero-order statistical model for DAC and AC, which is generated independently for each dataset. A 256-bit hash is used to detect decoding errors. We assume that transmission errors are resolved at a lower layer by means of retransmissions and/or forward error correction. The latter is common practice for satellite communication in general and nano satellites in particular [87]. Thus, the savings in transmission data achieved by our approach refer to transport layer goodput instead of raw channel capacity. A simultaneous treatment of transmission errors in the satellite downlinks is beyond the scope of this thesis.

In the following, we will explore the limits of how far we can reduce the rate of source X, so that it can still be successfully decoded given source Y as side information. Intentionally, we challenge our algorithms, in particular by increasing the overlap factor c so far that decoding will start to fail. We look at the number of datasets for which correct decoding is not achieved for different similarity functions and varying overlap factors. This shows what the safe parameter range is, and therefore how much additional compression (beyond standard arithmetic coding) can be achieved.

5.5.1 The Earth's Magnetic Field Data

The data obtained during the MagSat mission [137] serves as an example for a real-world application of DAC. Its goal was to obtain detailed data of Earth's magnetic field and thereby detect magnetic anomalies in the Earth's crust. The mission had a total duration of eight months, during which time the satellite circled Earth many times in low Earth orbit.

A single satellite was used in the MagSat mission. We used the fact that the satellite circled Earth repeatedly (about once every 94 minutes [137]) to simulate a nano-satellite formation based on the MagSat data. Among the MagSat data, there are pairs of datasets recorded by the *same* satellite on *different* overflights with a horizontal distance of less than 50 km at the equator. These were combined to simulate the measurement of Earth's magnetic field by two nano-satellites flying in close proximity.

The datasets are generated using the MagSat data, of which we considered only the magnetic field strength. The datasets typically consist of ca. 11,300 samples each, with a raw (uncompressed) size of ca. 23 kB. With arithmetic coding (i. e., without making use of

[4] For clarity, it should be noted that of course we still implement lossless compression, just not necessarily at the theoretically "perfect" Slepian-Wolf rate. Even if the resulting rate is higher than this limit, we do achieve rates better than what could be achieved by independently compressing both sources.

correlation across the two sources), we found that the size of a dataset can be reduced to typically 12 kB.

The data exhibit a high level of both temporal and spatial correlation. As can be seen in Fig. 5.1, the magnetic field exhibits low levels of noise and changes slowly compared to the sampling rate. Thus, the correlation of consecutive data points of each dataset, i. e., its autocorrelation, is high. The spatial correlation between the datasets obtained at different positions is high as well.

This scalar time series data is particularly suitable for our approach, as it allows for straightforward application of AC. Moreover, the partially decoded datasets represent completely decoded time series data on a sub-interval in time, thereby easing the design of appropriate fitness functions. DAC could in principle be applied to other kinds of correlated observation data, like overlapping photographic imagery. We leave that for future work, because it would require both a more complex encoding/decoding pipeline, as well as a plethora of additional degrees of freedom regarding possible fitness functions.

5.5.2 Evaluating the Decoding Performance

We used the adaptive decoder described above. Because the number of possible solution candidates grows exponentially with the number of bits saved, we stopped decoding when no solution was found before exceeding a limit of 1 GB RAM usage. The quality of the DAC algorithms was evaluated based on the number of correctly decoded datasets for different overlap factors.

Because the window size determines the length of the region in which the correlation is evaluated by the fitness function, its choice is very important for the fitness function's performance. To determine a suitable window size W for each fitness function, we use all 948 real-world datasets and a technique we call *fast decoding*. The idea of fast decoding is to decode each received dataset with the adaptive decoder, but accept any decoding path whose first α bits match the corresponding bits in the 256 bit hash of the correlated dataset. Because the correct 256 bit hash is known, we can then discern how many decoding results were correct. We observed that even if only the first $\alpha = 3$ bits were used, a large number of datasets was decoded successfully for a suitable W.

From a theoretical perspective, we would assume f's performance to be a concave function of W, because there are two competing effects associated with the window size. On the one hand, the larger the window size, the stronger the statistical significance of f's associated fitness value. On the other hand, the smaller the window size, the stronger the influence

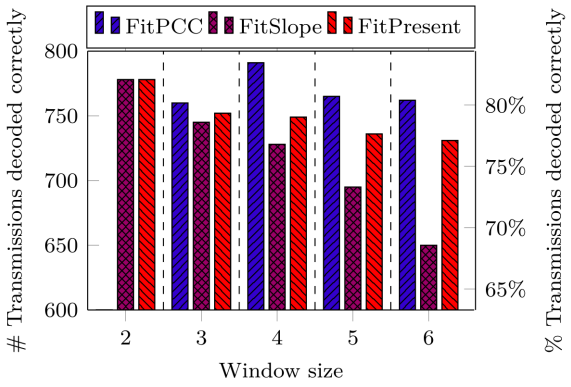


Fig. 5.2.: Fast decoding performance for the overlap factor $c = 1.1$ and different window sizes.

of the most recent path symbols on the fitness value, i. e., the sooner decoding errors can be detected and wrong decoding paths pruned.

Fig. 5.2 shows the results of such a fast decoding performance test with an overlap factor of 1.1. Note that only combinations of fitness functions and window sizes with more than 600 correctly decoded datasets are shown. As expected, the performance of FitPCC is a roughly concave function of W. For the other two fitness functions, this is not the case. Here, the optimal W is 2 for both functions, i. e., increasing the statistical significance seems to have less of an impact than detecting decoding errors early. Choosing a window size of 2 for both fitness functions is not meaningful in this specific case because for $W = 2$ FitSlope and FitPresent evaluate the same property. For this reason, we only used $W = 2$ for FitSlope, whose performance degrades strongest for bigger W, and $W = 4$ for FitPresent.

To investigate the achievable compression rate, we increased the overlap factor c stepwise and analyzed the number of datasets that could not be decoded before exceeding the resource limits.

Table 5.0 shows the decoding performance for an overlap factor of 1.2, which corresponds to a saving of about 296 bytes per dataset. FitPresent performs the worst with 41 undecoded datasets. As these 41 datasets correspond to $4.32\,\%$ of all datasets, we say that FitPresent's

Tab. 5.0.: Decoding performance for $c = 1.2$.

Fitness function	Window size	Not decodable	NDR
FitPCC	4	0	$0\,\%$
FitSlope	2	0	$0\,\%$
FitPresent	4	41	$4.32\,\%$

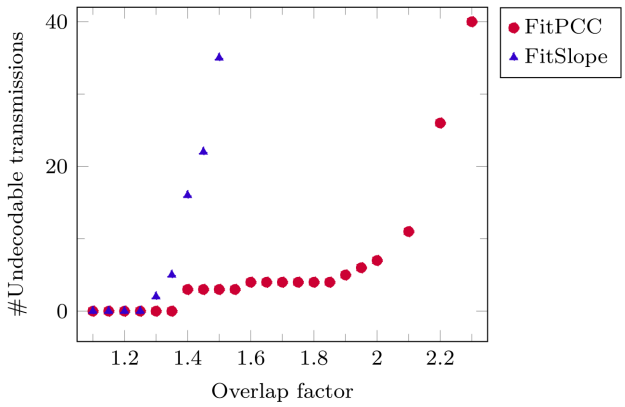


Fig. 5.3.: Number of undecodable transmissions using FitPCC or FitSlope.

non-decoding-rate (NDR) is $4.32\,\%$. FitPCC and FitSlope performed best and were able to decode *all* datasets for $c = 1.2$.

It is important to note that one of DAC's unique properties is that (in contrast to classical lossy compression algorithms) the correct dataset will theoretically always be decodable. In fact, there are many ways to recover missing transmissions, for example by changing the fitness function used for decoding or by running the same decoder on a computer with more main memory. Depending on the specific application setting, it is also well conceivable that a decoder that notices a decoding failure for a specific transmission requests additional data from the respective source using an interactive protocol.

The only hard limitation for the number of different decoding attempts for the same dataset is given by the number of hash comparisons that can be performed without risking a non-negligible false positive probability.

5.5.3 Exploring the Limits

So far, the two fitness functions FitPCC and FitSlope are the most promising of the five fitness functions we designed.

The results are depicted in Figure 5.3. When FitSlope is used and memory is limited to 1 GB, all dataset transmissions are decodable for an overlap factor of 1.25 or less. So FitSlope can be used for a size reduction of up to 0.322 bits per symbol, i. e., about 370 bytes per dataset, while still decoding all datasets successfully. For overlap factors greater than 1.25, however, the number of undecodable transmissions grows rapidly.

When FitPCC is used, all datasets can be decoded up to an overlap factor of $c = 1.35$, which is slightly better than FitSlope. The steep increase of undecodable datasets for

higher overlap factors starts much later than for FitSlope, but exhibits a qualitatively similar behavior. For FitPCC, either all datasets can be decoded when a size reduction of 0.433 bit per symbol (about 500 bytes per dataset) is sufficient, or up to 1 bit per symbol (about 1.2 kB per dataset) can be saved without raising the NDR above $0.75\,\%$. The latter means that the result is $10\,\%$ smaller than the dataset compressed with (non-distributed) arithmetic coding.

5.5.4 Discussion

The above results should be understood as a proof of principle, laying the groundwork for further research on applying DAC to real-world scenarios on real-world data. The fitness functions are not fully optimized and the results are not representative of what DAC could achieve if more realistic correlation models were used. Our intention was a proof of concept of a functional and realistic DAC algorithm for real-world scenarios.

Note that the rate reduction achieved by DAC considered in this work denotes the amount of bits saved per symbol *in addition* to what is already saved by using AC.

To keep decoding times manageable, we used only little resources for the evaluation of the decoding algorithms. In practice, faster CPUs and more main memory will be available for decoding, especially in our use case. This has the potential to decrease the NDR when compared to our results. The fact that the fitness function, i. e., an important part of the decoder, can be exchanged without needing to change the encoder, is a quite unique feature of DAC and provides an unusual amount of flexibility.

A key part of a functioning DAC algorithm is a suitable fitness function. In this work, we used the smoothness or shape of the curve to identify the correct decoding path and used a simple statistical model for AC. If a better statistical model were used for AC, the differences between correct and incorrect decoding paths could become more subtle.

Finally, though finding DAC algorithms with good performance is challenging, we believe it is currently promising to use distributed source coding under realistic assumptions.

5.6 Conclusion

The central question addressed in this chapter is whether and how DSC can be used to improve nano-satellite downlink efficiency, without increasing resource utilization on the satellites. We investigated a scenario in which two satellites measure Earth's magnetic field in low Earth orbit. Based on measured complex real-world data we demonstrated

that using only small realistic knowledge about the datasets' correlation, DSC techniques can be used to significantly reduce the amount of channel capacity used. Using a simple general-purpose encoding suitable for any kind of scalar time-series measurement data, our adaptive decoder together with suitable fitness functions is able to decode the rate-reduced transmitted data by exploitation of inter-dataset redundancy. Because compression has the potential to reduce both the energy demand for transmission *and* the memory needed for storage, DSC is an attractive approach for this kind of application. The proposed application of DSC is not an alternative to other lossless compression algorithms, but rather an additional layer of compression that allows (in scenarios where AC is already employed for compression) for further reduction of code word size on top of what traditional lossless codecs like AC achieve.

To the best of our knowledge, we are the first who successfully applied realistic DAC algorithms to a complex, distributed real-world setting and provided an extensive evaluation of its performance (based on more than 400 correlated datasets). We designed a new decoder for DAC, which adaptively chooses the pruning threshold. Furthermore, we proposed three fitness functions that avoid using detailed knowledge about probability distributions, and introduced a method to quickly determine the fitness functions' window sizes.

Finally, we evaluated the performance of our DAC decoder and the fitness functions for a wide range of overlap factors and provided a thorough discussion of our results.

Part II

Transmitting Bulk Data to Orbit

6 EAGER Decoding: Introducing EAger Gaussian Elimination for Rlnc Decoding

> **General:**
> *This is good, but what is best in life?*
> **Soldier:**
> *The open steppe, fleet horse, falcons at your wrist, and the wind in your hair.*
> **General:**
> *WRONG! Conan! What is best in life?*
> **Conan:**
> *To crush your enemies, see them driven before you, and to hear the lamentations of their women.*
>
> — **Conan the Barbarian**
> (1982)

TL;DR *When using Gaussian elimination for incremental RLNC decoding, eager back-substitution during injection is better than incremental LU decomposition with lazy back-substitution.*

acronym

6.0 Introduction

In random linear network coding (RLNC) over finite fields, decoding is often performed using Gaussian elimination (GE). While it would be possible for an RLNC-decoder using a generation size M to passively collect M linear combinations (LCs) before trying to solve the resulting system of linear equations, it has advantages to perform the GE incrementally each time a new LC is injected into, i. e., received by, the decoder: during incremental decoding, the decoder finds whether a newly injected linear combination is *innovative*, i. e., whether it is linearly independent of the linear combinations injected before. Thus, it

is able to keep track of the rank of the matrix formed by all injected linear combinations, it can use less memory by storing only innovative linear combinations, and it can randomly recode the received linear combinations using less computational resources if it computes new combinations only from innovatively injected LCs instead of from all injected LCs.

Many of the RLNC-based coding schemes described in the literature therefore decode packets incrementally to some degree, but what and how much decoding work is actually done at what time differs significantly.

In this chapter we introduce EAGER[5], an approach to decode LCs more eagerly than proposed in the related literature. We show that our approach, which is applicable to creating variants of many known coding schemes, comes with asymptotically negligible computational overhead or even improves performance, depending on which existing scheme it is applied to. At the same time it reduces the decoder-induced overhead as well as network-induced overhead, again depending on which scheme our approach is applied to. In the context of layered prioritized RLNC, it supersedes computationally expensive techniques like counter-elimination [88], as our evaluation shows. Even for decoders handling bulk encoding matrices, it reduces the computational demand to decide whether an LCs is innovative. Thus, the overall computational complexity of decoding is reduced. Most importantly, EAGER is able to bring its decoder, after each injection, to a normalized state that uniquely depends on the span of the LCs injected, requiring only marginal computational overhead. Thereby, it enables efficiently determining the equality of two decoders by exchanging only hash values between the corresponding nodes. The consequences thereof are discussed in detail in Chapter 7.

The rest of this chapter is structured as follows: in Section 6.1 we discuss the related work, followed by Section 6.2 where we recapitulate GE and LU decomposition in the variants that are used for RLNC decoding in the related work. EAGER decoding itself and its data structures are introduced in Section 6.3 and an analytical treatment of its computational resource utilization in comparison with LU decomposition is given in Section 6.4. In Section 6.5 we show how EAGER's engine can be applied to different sparsity classes, including a comparative empirical evaluation in Subsection 6.5.1 in the context of layered prioritized RLNC. In Section 6.6 we discuss how EAGER can relatively simple be extended to maintain a normalized decoder state that depends only on the subspace spanned by the LCs injected. We also measure the computational cost of maintaining this state empirically in Subsection 6.6.2. Finally, we draw a short conclusion in Section 6.7.

Regarding our algorithms name we clarify here that EAGER is called *eager* as in "lazy vs. eager evaluation" but it is not *greedy* as in "greedy vs. thrifty optimization".

[5]The expansion of the acronym EAGER can be found in this chapter's title.

6.1 Related Work

The key idea of our approach is to replace the usual row echelon form that is central to solving linear systems using GE by the *bilateral row* echelon form. Therefore, this work is primarily related to RLNC schemes using some flavor of LU decomposition and/or GE for decoding and less related to LT Codes [89] or Raptor Codes [90], both of which are fountain codes [91] using random LCs as code symbols, but whose encoding vector distributions allow for more efficient decoding algorithms than GE-based approaches. For LT Codes it has been shown that in-network recoding is also applicable [92] to some degree.

Many known GE-based RLNC de- and recoding approaches already are somewhat eager in the sense that they perform the LU decomposition part of decoding *incrementally* (also termed *on-the-fly* by some authors [93]) in order to decide for each incoming LC whether it is innovative, i. e., whether it is linearly independent of the set of LCs injected so far. Incremental decoding therefore means that the encoding vectors of received LCs are not simply buffered, but are immediately processed using the current decoder state and then stored in some kind of row echelon form.

Some authors favor laying out their decoder in LU factorized form with partial pivoting [88], meaning that the encoding vectors of incoming LCs are stored in decomposed form in triangular matrices L and U (the latter being in row echelon form) while the information vectors (IVs) are simply buffered together with a permutation matrix. The forward and backward substitution part needed for decoding are then *lazily* postponed until the decoder state LU is of full rank.

Others [94, 93] eagerly perform forward substitution incrementally as well, which not only releases from the necessity to store L in memory but also increases the opportunities of recoding LCs that satisfy certain sparsity conditions. In addition, performing the forward substitution on-the-fly when packets are received, implies that the decoder literally stores non-trivial linear recombinations of received LCs. Transmitting these partially decoded LCs already constitutes recoding while avoiding the computational demand of explicitly creating new LCs for transmission, a technique known as *precoding* [95, 96].

In [97] the concept of overlapping generations is introduced, but a separate decoder for each generation is used. Every time a generation is completely decoded, a number of LCs is back-substituted into adjacent generations. Our decoder EAGER, by contrast, can be used to jointly decode the overlapping generations as one system of linear equations, thereby reducing decoding overhead.

We think that EAGER decoding could be used for the decoding of “chains of chunks” introduced in [98] as well. The technique of “combination of chunks” introduced there

means to jointly decode contiguous undecodable overlapping chunks using one joint decoder, something that EAGER is particularly well-suited for.

To the best of our knowledge, none of the chunked or layered RLNC-based codes described in literature uses incremental backward substitution in decoding in an eager per-linear-combination way, as our proposed decoding method does. Apart from improving some codes described here in terms of network-induced and decoder-induced overhead as well as computational costs, we see the main contribution of this chapter in providing a simple decoder layout that is suitable for implementing decoding and recoding for most of the mentioned codes in a unified way. In many cases our approach removes the necessity to handle transmissions consisting of many generations/chunks/layers by a bunch of interacting sub-decoders.

Finally, we did not find any reference in literature to an RLNC decoder whose state depends uniquely on its current subspace. In Section 6.6.1 we show that EAGER can be modified to gain that property and Chapter 7 is dedicated to demonstrate by example of over-the-air programming of satellite formations how RLNC-based protocols can benefit from this feature. However, we also did not find any explicit reference supporting our assumption that no state-of-the-art decoder is capable of maintaining a unique decoder state.

6.2 RLNC Decoding Using Gaussian Elimination

6.2.0 Problem Statement

Consider a random linear code, i. e., a code in which code symbols consist mainly of random LCs of vectors over a finite field $\mathbb{F}_q$. q is the size of the finite field and usually chosen to be an integer power of 2. Let X be an $M \times K$ matrix over $\mathbb{F}_q$ that represents the plain uncoded *bulk payload* that is to be encoded. Some authors call this the "data object" [99] or "long file" [100]. In erasure code literature, the concept of one bulk payload is often completely reduced to the notion of a "set of source symbols" or "original packets". The term "bulk payload" that we use corresponds to the data that is encoded by the whole set of original packets.

Then every pair of encoding vector (EV) $a \in \mathbb{F}_q^M$ and information vector (IV) $b \in \mathbb{F}_q^K$ satisfying $a \cdot X = b$ is called a *code word* or a *LC*.

Encoding A source node (also referred to as *encoder*) can then create an arbitrary number of code words using the following steps:

0. Draw a random *Encoding vector* $\mathbf{a}_i = (a_{i,0}, \dots, a_{i,M-1}) \in \mathbb{F}_q^M$. The discrete random distribution used to draw the EV heavily depends on the specific code used, might enforce certain sparsity pattern in $\mathbf{a}_i$ (see paragraph *Sparsity of encoding vectors* on p. 100), and may also depend on additional feedback that the source received from other nodes.

1. Calculate the *Information vector* $\mathbf{b}_i$ as the linear combination of the rows of X given by (6.0).
$$b_{i,j} = \sum_{k=0}^{M-1} a_{i,k} X_{k,j} \tag{6.0}$$
The pair (EV, IV) will be referred to as *linear combination (LC)*.

2. Create a *coded packet* that enables a receiving node to restore (EV, IV). The most straightforward choice would be to use the concatenation of the bit representations of EV and IV as coded packet. A reasonable choice in scenarios of end-to-end fountain coding without in-network recoding would be to construct the EV using a suitable hash function applied to i and use (i, IV) as the coded packet. For present purposes, it is not important how coded packets are serialized. Instead, it is sufficient to assume that when a node receives a packet, the full LC is injected into the decoder.

Decoding In order to obtain X, a receiving node can arrange the LCs encoded in the received coded packets in matrices A, whose rows are the EVs, and B, whose rows are the corresponding IVs, and solve the resulting system of linear equations $AX = B$ for X. Solving is of course only possible if enough packets have been received, such that $\operatorname{rank} A = M$.

In addition, it is often desired (see Section 7.2.3) that the decoder can always be queried for the rank of the matrix whose rows are the encoding vectors of all packets injected so far. Exchanging this information can, for example, help to estimate whether coded packets generated by one node are likely to be innovative for another node.

This ability to report the decoder's rank is of course often also desired with respect to ranks of sub-matrices corresponding to chunks or layers, when applicable. In general, by a "decoder's rank" we mean the dimension of the linear span of the LCs injected into the decoder.

Recoding Following the nomenclature in [95], recoding is the process of computing new LCs by linearly combining the LCs of received packets, which includes the combination of the corresponding IVs as well as the received EVs. Computing new LCs from a completely decoded data-set, however, is essentially identical to encoding and therefore termed *re-encoding*, when performed by a node not being an initial source node.

Sparsity of encoding vectors Many random linear codes restrict the encoding vectors to certain sparsity patterns. Let $\mathcal{C} \subseteq \mathcal{P}([0, M)_{\mathbb{Z}})$ be a set of *sparsity classes*. Note that the term *class* is used by some authors, e. g., [97], for what we call a *generation*. An encoding vector $\mathbf{a}_i$ is said to belong to a class $C \in \mathcal{C}$ if $\{j \colon a_{i,j} \neq 0\} \subseteq C$. The classes do not need to be disjoint, so an EV can belong to more than one class. For each packet to be encoded, a source node selects a sparsity class from $C \in \mathcal{C}$ and then draws a random EV belonging to C. The process of selecting C differs across codes and may depend on the source's knowledge about the other nodes' state. Examples for different sparsity class approaches are depicted in Fig. 6.0. All of these examples share the property that all sparsity classes are integer intervals shape, i. e., every class $C \in \mathcal{C}$ can be represented as:

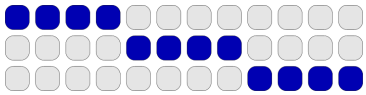
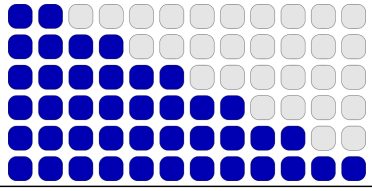
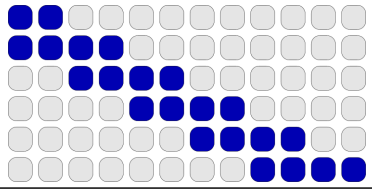
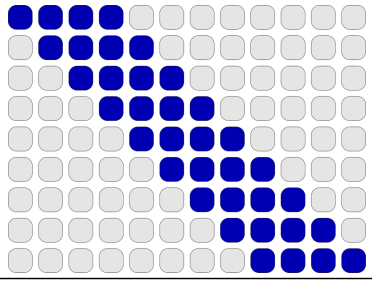
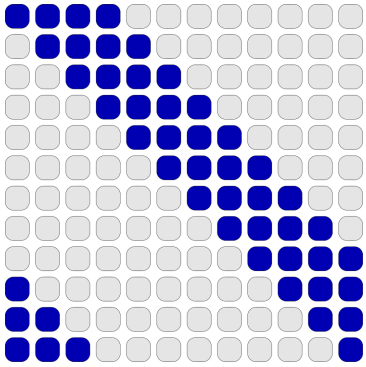
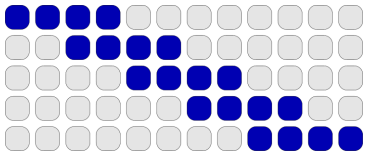
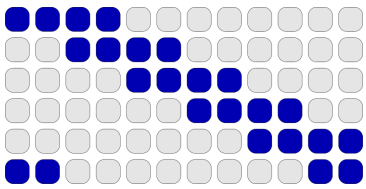
$$C = [t, t + \ell)_{\mathbb{Z}} \tag{6.1}$$

or

$$C = [t, M)_{\mathbb{Z}} \cup [0, t + \ell - M)_{\mathbb{Z}} \tag{6.2}$$

We therefore call these codes *interval-sparse codes*. If an interval-sparse code cannot be represented without classes of the form (6.2), we call it a *wrapped interval-sparse code*. We emphasize here that for a random linear *network* code it is important that intermediate nodes are able to respect these sparsity classes when recoding packets. For a recoding node, it is possible to generate a packet belonging to a class C by simply building a random linear combination of all received innovative LCs belonging to C. This naïve strategy, however, misses some recoding possibilities, because the number of innovative LCs belonging to C that have been received by a node may be significantly smaller than the dimension of the decoder's state with respect to C. As a minimal non-trivial example, consider a code with $M = 4$ and $\mathcal{C} = \{\{0, 1, 2\}, \{1, 2, 3\}\}$, and a decoder that has received two independent LCs belonging to $C_0 = \{0, 1, 2\}$ only:

$$\mathbf{a}_0 = (1, 1, 1, 0)$$
$$\mathbf{a}_1 = (1, 2, 2, 0)$$

RLNC The base-case of one single large generation.	
chunked RLNC Multiple generations can be interpreted as one large generation divided into chunks by means of disjoint sparsity classes.	
Bulk HNC The sparsity classes used in [101].	
Band HNC Like Bulk HNC, but combined with the idea of band codes; used in our evaluation.	
Continuous band code; not wrapped	
Continuous band code; wrapped	
Overlapped chunked code; not wrapped	
Overlapped chunked code; wrapped	

Tab. 6.0.: Examples of sparsity classes of different codes.

Since $\mathbf{a}_0 \oplus \mathbf{a}_1 = (0, 3, 3, 0)$ belongs to $C_1 = \{1, 2, 3\}$ as well, the decoder state with respect to C_1 has a dimension of 1, despite having received no LC belonging to C_1 at all. We term this kind of recoding *inter-class recoding*.

6.2.1 Solving Systems of Linear Equations Using Gaussian Elimination with Row Pivoting

We start with a recap of traditional numerical solving of floating-point systems of linear equations through GE-based LU decomposition with lazy forward and backward substitution. Then we discuss how and in how far the steps of row-pivoting, GE, and forward substitution are handled differently by the related work.

Finally, we introduce the notion of *bilateral row echelon form*, a novel criterion for matrices representing the LHS decoder state that can be achieved and efficiently retained using eager incremental backward substitution without significant computational overhead.

Vanilla numerical substitution-after-column-by-column-decomposition

The standard numerical method to solve a system of linear equations $AX = B$ with non-singular $(M \times M)$-matrix A and right-hand side B for X, e. g., as implemented in LAPACK's `dgesv` routine [138], consists of the following consecutive steps:

0. Factor $A = PLU$ using GE, where
   - P is a permutation matrix (implemented as array of integers) that represents the row swaps needed for pivoting,
   - L is a lower unitriangular bulk matrix,
   - U is an upper triangular matrix,
1. Compute $Y = L^{-1}P^{-1}B$, also known as forward substitution.
2. Compute $X = U^{-1}Y$, also known as backward substitution.

LU decomposition is performed through GE, creating the zeros below the diagonal column by column, because row pivoting, used for sake of numerical stability, means to use the largest-by-absolute-value element of the corresponding sub-column to eliminate the remaining non-zero elements within that sub-column. In order to compute U, one initializes $U^{(0)} = A$, which is completely known by then, and in each step of the outermost loop GE is applied to eliminate all below-diagonal non-zero elements of one column, so that after i steps, the sub-matrix consisting of the i leftmost columns of $U^{(i)}$ is in row echelon form.

Definition 6.1 (Head and tail index)**.** For $a \in \mathbb{F}_q^n$ we define the head and tail index as the (inclusive) boundary indices of the convex hull of non-zero elements of a:

$$h(a) := \begin{cases} n & \text{if } a \equiv 0 \\ \min\{i \in [0,n)_\mathbb{Z} \mid a_i \neq 0\} & \text{else} \end{cases}$$

$$t(a) := \begin{cases} -1 & \text{if } a \equiv 0 \\ \max\{i \in [0,n)_\mathbb{Z} \mid a_i \neq 0\} & \text{else} \end{cases}$$

By $a \equiv 0$ we mean that every element of a equals 0.

Definition 6.2 (Row echelon form)**.** Matrix $A \in \mathbb{F}_q^{m \times n}$ is in row echelon form iff $\exists k \in [0,m]_\mathbb{Z}$:

$$\begin{aligned} & A_i \equiv 0 \forall i \in [k,m)_\mathbb{Z} \\ \wedge\ & h(A_i) \in [0,n)_\mathbb{Z} \forall i \in [0,k)_\mathbb{Z} \\ \wedge\ & h(A_i) < h(A_j) \forall i,j \in [0,k)_\mathbb{Z} \mid 0 \leq i < j < k \end{aligned}$$

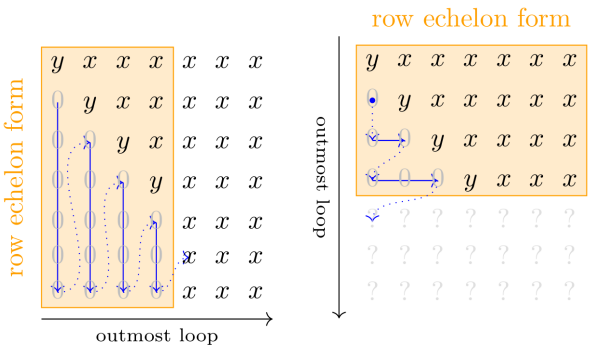
Where A_i refers to the i^{th} row of matrix A.

Incremental decomposition

For decoding in RLNC, this column-by-column, left-to-right strategy is neither compatible with the demand for incremental processing of LCs (i. e., rows), nor is it needed in systems over finite fields, where field arithmetic is exact as opposed to floating point arithmetic that is usually used in a numerical context.

For this reason, it is common (as described in [102] and in more detail in [88]) in RLNC decoding, to factorize A row by row, such that after i steps, the sub-matrix consisting of the i topmost rows of $U^{(i)}$ is in row echelon form. Note however that these first i steps may correspond to injecting $i' >= i$ LCs with a total rank of i: transforming i' LCs that span a subspace of rank i into row echelon form produces $i' - i$ all-zero rows (representing the non-innovative packets) below the i non-zero rows. Since the all-zero rows carry no information, they are simply neglected. These different eliminations strategies are visualized in Fig. 6.0.

Besides the lack of necessity to take numerical stability into account, which enables factoring A row by row, there is yet another important difference between floating-point and RLNC LU decomposition: in numerics, bulk matrices, e. g., A and B, are often represented by storing their elements consecutively in memory, let it be in Fortran- or C-order. When the problem of solving $AX = B$ is separated into LU decomposition and



(a) Vanilla numerical GE. **(b)** Incremental decoding.

Fig. 6.0.: Comparison of (a) vanilla floating point LU decomposition with row-pivoting and (b) incremental decoding. The "?"s represent zeros corresponding to yet unknown LCs. The meta variables refer to arbitrary non-zero elements (y) and arbitrary elements (x) respectively, i. e., the instances of x and y do not imply equality. The highlighted submatrices are in row echelon form and their elements are *final*, i. e., are not altered by subsequent iterations. For incremental decoding (b) the figure depicts only the simplest case in which row-pivoting is unnecessary.

substitution steps, implemented in different subroutines, it is required to keep track of the permutation P that allows for stable decomposition. But even if the decomposition routine had access to B, swapping the actual full rows of U^i and B in-place during decomposition would be wasteful.[6]

In RLNC decoding we are in a fundamentally different position, in that A and B are not fully known while we decompose A. When a new innovative LC is injected into a decoder of rank i, there is no good reason to store the EV and IV in the first vacant row of A and B respectively (as it is done in [102]), apart from the abstract notion of *row echelon form* of U.

In-place Permutation

A different strategy, e. g., used to decode Band Codes [94] (RLNC over $\mathbb{F}_2$ with band-diagonal encoding matrices), is to keep U in a state where each row $U_j^{(i)}$ is either vacant or has exactly j leading zeros. When swapping rows, instead of explicitly recording the swap history in a dedicated data structure P, they just simultaneously swap pointers in their representation of U and B. When neglecting the concept of column-wrap-around,

[6] An implementer not tied to historical bounds of existing libraries could of course abandon continuous memory layout for bulk matrices and instead represent a matrix as an array of pointers to row vectors. But computationally that would not make any real difference to the status quo, where a permutation is implemented as a one-dimensional array of integers that itself *is an array of pointers*, just not of machine-address-space pointers, but row-index-space pointers.

the same technique is used in [93]. We shall note here that although this approach seems to make matrix U deviate from classical row echelon form, the sub-matrix that consists of the non-vacant rows of U still is maintained in row echelon form. We therefore term this condition *spread row echelon form*.

Definition 6.3 (Spread row echelon form)**.** Matrix $A \in \mathbb{F}_q^{m \times n}$ is in spread row echelon form iff

$$\forall i \in [0, m)_{\mathbb{Z}} \colon A_i \equiv 0 \vee h(A_i) = i$$

Essentially, a matrix in row echelon form can be transformed to spread row echelon form by placing every non-zero row A_i at row $h(A_i)$.

Again, it is up to the implementer's taste whether the matrix itself is organized in this form, or an additional index structure is used to find a row with a given number of leading zeros in constant time.

Eager Forward Substitution

Another concept used in [94, 93], even though not termed that way, is what we call *eager forward substitution*. By this we mean that row operations applied on $U^{(i)}$ are directly applied on B as well, instead of saving them in a column of L. It turns out that the decomposition-approach of lazily recording row operations in a matrix L and postponing forward substitution $Y := L^{-1}B$ until the decoder is of full rank, costs exactly as many elementary multiplications, so that eager forward substitution has no computational overhead, with two exceptions:

(a) It *is worth* to postpone the row operations on B until it is found that the injected LC is innovative, as it is discarded anyway otherwise.

(b) The computational effort of row operations on B is of course wasted, if the whole transmission is discarded before it is completed.

From now on, we will use a slightly different terminology: let G be an $(M \times M)$ *decoder matrix* and Y an $(M \times K)$ *data matrix*, both of which shall from now on be thought as mutable matrices which relieves us of carrying the upper indices like in $U^{(i)}$ through all calculations. Initially both G and Y shall be all-zero. As LCs are received and decoded iteratively, they are inserted into vacant rows of G and Y and both matrices are simultaneously manipulated using elementary row operations. Thus, at any stage of decoding $GX = Y$ will hold. If we speak of the decoder's rank, we mean the row rank of G.

6.3 EAGER Decoding

6.3.0 Eager Backward Substitution

Up to here we only gave a wrap-up of related work and introduced some notation. Now we start with our own contribution: the eager backward substitution.

The idea is simply not to limit incremental decoding to the elimination of non-zeros in the decoder matrix, until no pair of rows has the same number of leading zeros (i. e., it is in row echelon form), but to also eliminate non-zeros at the rear end of rows, until no pair of rows has the same number of trailing zeros.

We could have defined the eager-forward-substitution strategy equivalently as follows: "When injecting a new EV into the decoder, place it at a vacant row of G and use elementary row operations until G is in spread row echelon form. If we find that the LC is innovative, also place the IV in Y and apply the same sequence of row operations on Y." The step of eager backward substitution then reads: "Then, use elementary row operations until G is in bilateral spread row echelon form (Definition 6.5). Also apply each operation on Y." Examples for the sparsity patterns of decoder matrices in the different types of row echelon forms are given in Fig. 6.1.

Definition 6.4 (Ordered head- and tail-unique basis (OHTUB))**.** An OHTUB in $\mathbb{F}_q^n$ is a (potentially empty) tuple B of vectors $\in \mathbb{F}_q^n$ satisfying

(α) $b \not\equiv 0 \forall b \in B$

(β) $h(B_i) < h(B_j) \forall i, j \in [0, |B|)_{\mathbb{Z}}, i < j$ (head-ordered and -unique)

(γ) $t(B_i) \neq t(B_j) \forall i, j \in [0, |B|)_{\mathbb{Z}}, i \neq j$ (tail-unique)

Of course, for $|B| < n$, B is not a base of $\mathbb{F}_q^n$. However, the elements of B are linearly independent as stated in Corollary 6.2.

Proposition 6.1. *Given an OHTUB B and a non-zero vector $a \in \mathbb{F}_q^n$:*

$$h(a) \neq h(b) \forall b \in B \Rightarrow a \notin \operatorname{span} B$$

Where span B *denotes the linear span of the rows of a matrix B.*

Proof. We proof the contraposition. $a \in \operatorname{span} B \Rightarrow \exists \lambda \in \mathbb{F}_q^{|B|} \colon a = \sum_{i=0}^{|B|-1} \lambda_i B_i$.
$a \not\equiv 0 \Rightarrow \exists i \colon \lambda_i \neq 0$. Let $j := \min\{i \in [0, |B|)_{\mathbb{Z}} \mid \lambda_i \neq 0\}$. $\Rightarrow h(a) = h(B_j)$. □

Corollary 6.2. *The elements of an OHTUB are linearly independent.*

vanilla
row echelon form

$$\begin{matrix} y & x & x & x & x & x & x \\ 0 & y & x & x & x & x & x \\ 0 & 0 & y & x & x & x & x \\ 0 & 0 & 0 & y & x & x & x \\ ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? \end{matrix}$$

(a) $G^{(\alpha)}$ in row echelon form (REF)/spread row echelon form (SREF).

spread
row echelon form

bilateral spread
row echelon form

$$\begin{matrix} y & x & x & y & 0 & 0 & 0 \\ 0 & y & x & x & y & 0 & 0 \\ 0 & 0 & y & x & x & y & 0 \\ 0 & 0 & 0 & y & x & x & y \\ ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? \end{matrix}$$

(b) $G^{(\alpha)}$ in bilateral spread row echelon form (BSREF).

$$\begin{matrix} y & x & x & x & x & x & x \\ 0 & 0 & y & x & x & x & x \\ 0 & 0 & 0 & y & x & x & x \\ 0 & 0 & 0 & 0 & 0 & y & x \\ ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? \end{matrix}$$

(c) $G^{(\beta)}$ in REF.

$$\begin{matrix} y & x & x & x & x & x & x \\ ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & y & x & x & x & x \\ 0 & 0 & 0 & y & x & x & x \\ ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & 0 & y & x \\ ? & ? & ? & ? & ? & ? & ? \end{matrix}$$

(d) $G^{(\beta)}$ in SREF.

$$\begin{matrix} y & x & x & y & 0 & 0 & 0 \\ ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & y & x & y & 0 & 0 \\ 0 & 0 & 0 & y & x & x & y \\ ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & 0 & y & 0 \\ ? & ? & ? & ? & ? & ? & ? \end{matrix}$$

(e) $G^{(\beta)}$ in BSREF.

Fig. 6.1.: Schematic of different encoder matrixes $G^{(\alpha)}$ (a, b) and $G^{(\beta)}$ (c,d,e) in different row echelon forms. Each encoder matrix has rank 4. These sparsity patterns are only two possible examples and depend on the specific values of x and y. E. g., the shape of (e) requires that (d): $G^{\beta}_{56} = 0$.

Proof. Follows from Proposition 6.1 by induction. □

Proposition 6.3. *Given an OHTUB B and a non-zero vector $a \in \mathbb{F}_q^n$:*

$$(\exists i \in [0, |B|)_{\mathbb{Z}} : h(a) = h(B_i) \wedge t(a) < t(B_i)) \Rightarrow a \notin \operatorname{span} B$$

Proof. We proof the contraposition. $a \in \operatorname{span} B \Rightarrow \exists \lambda \in \mathbb{F}_q^{|B|} : a = \sum_{j=0}^{|B|-1} \lambda_j B_j$.
Let $i \in [0, |B|)_{\mathbb{Z}}$: $h(a) = h(B_i)$ which exists because of Proposition 6.1. $\Rightarrow \lambda_i \neq 0$. t is injective on $B \Rightarrow t(a) = \max\{t(B_j) \mid j \in [0, |B|)_{\mathbb{Z}} \wedge \lambda_j \neq 0\} \geq t(B_i)$. □

Definition 6.5 (Bilateral spread row echelon form)**.** Matrix $A \in \mathbb{F}_q^{m \times n}$ is in bilateral spread row echelon form iff all the following hold:

(α) A is in spread row echelon form.

(β) The non-zero rows of A form an OHTUB.

It is obvious that every matrix A in spread row echelon form can be brought to bilateral spread row echelon form using elementary row operations: while there exist two rows $i < j$ with $t(A_i) = t(A_j)$, we can eliminate the last non-zero element of row i with row j. This does not change $h(A_i)$, but decreases $t(A_i)$ at least by one. As the total number of trailing zeros of all rows is bounded by the matrix size, iterating this process will deterministically terminate. When it terminates, the desired condition is fulfilled.

In order to perform the rear-end elimination efficiently, that is, to find a pair of rows with equal number of trailing zeros, we use a lookup table $t^{\text{inv}} \in \left([0, M)_{\mathbb{Z}} \cup \{\emptyset\}\right)^M$ to partially invert the tail index function t:

$$t_i^{\text{inv}} := \begin{cases} j & \text{if } t(G_j) = i \\ \emptyset & \text{if } \nexists j \in [0, M)_{\mathbb{Z}} : t(G_j) = i \end{cases} \tag{6.3}$$

When a new row is injected and eliminated at the front (row echelon form), we can ripple-eliminate at the rear-end side using t^{inv} until a vacant position is found, which means that the bilateral spread row echelon form is reached.

The eager decoder uses the following decoding state:

0. Matrices G and Y over $\mathbb{F}_q$, implemented such that replacing a row with a given vector, swapping rows, etc., can be done in constant time. Initially, G and Y are all zero.

1. Lookup tables $t \in \{-1, \dots, M-1\}, t_i = t(G_i)$ and t^{inv} according to (6.3). Initially $t^{\text{inv}} \equiv \emptyset$ and $t \equiv -1$.

Pseudocode of the EAGER decoding is given in Algorithm 6.0. Note however that depending on the specific code EAGER is applied to, it can be favorable to replace the lazy for operations on Y (lines 4, 11) by eagerly applying them in line 4. While doing so comes at the expense of additional cost to check injected LCs for innovativeness, it is required to perform the swapping feature in Swap Gaussian Elimination (SGE) which plays a significant role in precoding, where this additional cost is compensated by a reduced cost for generating new LCs for recoding.

```
0  function INJECT(encoding vector g, information vector y)
1      l ← ⟨empty list⟩                          ▷ buffer to record elimination steps
2      s ← h(g)
3      while G_s ≠ ∅ do
4          append (s, g_s / G_ss) to l           ▷ record elimination step
5          eliminate g using G_s                 ▷ eliminate
6          if g ≡ 0 then
7              return False                      ▷ (a, b) is not innovative
8          s ← h(g)
                                                 ▷ G is not altered yet.
9      G_s ← g                                   ▷ insert g in G
10     for (r, x) ∈ l do
11         y ← y − xY_r                          ▷ replay eliminations l on y using rows of Y
12     Y_s ← y                                   ▷ insert y in Y
                                                 ▷ G is in non-bilateral spread row echelon form.
13     while t^inv_{t(G_s)} ≠ ∅ do
14         (s, r) ← (min{s, ρ}, max{s, ρ}) where ρ = t^inv_{t(G_s)}
15         rear-end eliminate G_s using G_r
16         apply the same row operation on Y
17         t^inv_{t(G_r)} ← r
18     t^inv_{t(G_s)} ← s
                                                 ▷ G is in bilateral spread row echelon form again.
19     return True
```

Listing 6.0.: High-level generic pseudocode of EAGER.

Proposition 6.4. *Given two matrices $G, H \in \mathbb{F}_q^{M\times M}$ in BSREF, it holds that*

$$(\alpha) \qquad \operatorname{span} G = \operatorname{span} H \Rightarrow h(G_i) = h(H_i) \forall i \in [0, M)_{\mathbb{Z}}$$

$$(\beta) \qquad \operatorname{span} G = \operatorname{span} H \Rightarrow t(G_i) = t(H_i) \forall i \in [0, M)_{\mathbb{Z}}$$

Proof. Both are proven by contraposition.

(α) $\exists i \in [0, M)_{\mathbb{Z}} \colon h(G_i) \neq h(H_i) \Rightarrow (h(G_i) = M \wedge h(H_i) = i) \vee (h(G_i) = i \wedge h(H_i) = M)$. WLOG $h(G_i) = i \wedge h(H_i) = M \Rightarrow G_i \notin \operatorname{span} H$ because of Proposition 6.1 $\Rightarrow \operatorname{span} G \neq \operatorname{span} H$. □

(β) $\exists i \in [0, M)_{\mathbb{Z}} \colon t(G_i) \neq t(H_i)$. If $G_i \equiv 0$ or $H_i \equiv 0$, then $\operatorname{span} G \neq \operatorname{span} H$ because of Proposition 6.1. Else, WLOG let $t(G_i) < t(H_i) \Rightarrow G_i \notin \operatorname{span} H$ because of Proposition 6.3. $\Rightarrow \operatorname{span} G \neq \operatorname{span} H$. □

Corollary 6.5. *Given a matrix G in BSREF, if G has full rank, then G is diagonal.*

Proof. $\operatorname{rank} G = M \Rightarrow \operatorname{span} G = \mathbb{F}_q^M = \operatorname{span} 1$. 1 is diagonal $\Rightarrow$ by Proposition 6.4 G is diagonal. □

Proposition 6.6. *Given a finite sequence L of LCs, if there exists an algorithm that is capable of partially decoding L, the same partial result will naturally be produced by an EAGER decoder when injecting all elements of L.*

Proof. Before sketching a proof, we need to clarify what we mean by "partially decode": by completely solving $AX = B$ we mean that we compute X, i. e., every row of X. By partially solving $AX = B$ for row $i \in [0, n)_{\mathbb{Z}}$ we mean that we compute X_i, regardless of whether the rest of X can also be computed yet. Any algorithm can only compute X_i from L if the standard basis vector $\mathbf{e}_i$ lies in the span of L's EVs. After injecting every LC of L into an EAGER decoder, its $\operatorname{span} G$ equals the span of L's EVs. $\Rightarrow \mathbf{e}_i \in \operatorname{span} G \Rightarrow h(G_i) = t(G_i) = i$ because of Proposition 6.3. $\Rightarrow X_i = (G_{ii})^{-1} Y_i$. □

6.3.1 Conditional Row Swapping

As already discussed, decoding algorithms of codes like SGE perform row swapping during the GE phase: before eliminating g using $G_{h(g)}$, g and $G_{h(g)}$ are always swapped (as well as y and $Y_{h(g)}$) which enables treating the rows of G and Y as *precoded LCs*. It turns out that even without precoding, this kind of swapping can save many rear-end elimination steps. When an LC (g, y) with a small tail index is eliminated using a row (G_i, Y_i) with a greater tail index ($t(g) < t(G_i)$), both LCs have that larger tail index $t(G_i)$ afterwards. Since the goal of eager back-substitution is to reduce the tail indices of G, some of this work can be saved by swapping the LCs before elimination. At a first glance, this seems to be in conflict with the strategy of postponing the GE row operations on Y until the innovativeness of g has been shown. But $t(g) < t(G_i) \Rightarrow g \notin \operatorname{span} G$ (Proposition 6.3), so as soon as we encounter a situation where (g, y) and (G_i, Y_i) *should* be swapped, we

know that (g, y) is innovative. Therefore, we can safely apply on y the postponed row operations recorded in l before swapping. From this point where we know that (g, y) is innovative, every row operation on g can immediately be applied on y as well.

As a final note, we want to emphasize that conditional row swapping is only an improvement of the computational performance and does not alter the partial decoding properties in any way, i. e., it is not required for zero-overhead early partial decoding in a Hierarchical Network Coding (HNC) context. Pseudocode of the full EAGER decoding including conditional row swapping is given in Algorithm 6.1.

6.4 Computational Demand

The computational demand for decoding strongly depends on the sparsity of injected LCs, which in turn differs for the specific codes. At first, we consider the computational demand of decoding a bulk code where all EVs are uniformly drawn from $\mathbb{F}_q^M$. We compute the worst case where each GE eliminates only one single element to zero which approximately corresponds to the limit of large finite field size q (for random rows, the probability that an elimination step produces more than one leading zero is roughly $\frac{1}{q}$). Furthermore, we restrict this calculation to decoding of exactly M innovative LCs, thereby neglecting the cost of eliminating any non-innovative LCs' EV to zero. In order to compare EAGER decoding with row-pivotized incremental LU decomposition, we define the computational demand of decoding as the number of $\mathbb{F}_q$ multiplications needed to compute X, thereby neglecting $\mathbb{F}_q$ additions (which are just bitwise XOR for $\log_2 q \in \mathbb{Z}$), modifications in the permutation matrix, pointer swaps, and so forth.

Both for LU decomposition and for EAGER, multiplications happen only during row operations. A row operation on Y simply needs K multiplications, because regardless of the sparsity of G (or U), Y is always bulk. Row operations on G and U, however, are not that simple, because in both G and U the strictly lower triangle equals zero by construction, thereby systematically reducing number of multiplications required for elimination. A similar argument holds for the trailing zeros in G.

6.4.0 LU Decomposition

Computational Demand of Decomposition

Assuming that no zeros occur by chance, we can neglect permutation and thus assume that after k LCs injected, $U_{ij} = 0 \forall j < i$, $U_{ii} \neq 0 \forall i < k$, and $U_{ij} = 0 \forall i \geq k \forall j$. To

```
0  function INJECT(encoding vector g, information vector y)
1      l ← ⟨empty list⟩                          ▷ buffer to record elimination steps
2      i ← False                                 ▷ i = whether g is definitely innovative
3      s ← h(g)
4      while G_s ≠ ∅ do
5          if t(g) < t(G_s) then
6              for (r, x) ∈ l do
7                  y ← y − xY_r                  ▷ replay eliminations l on d using rows of Y
8              i ← True
9              l ← ⟨empty list⟩
10             swap (g, y) and (G_i, Y_i)
11         if i then
12             y ← y − (g_s / G_ss) Y_s          ▷ perform elimination on y
13         else
14             append (s, g_s / G_ss) to l       ▷ record elimination step
15         eliminate g using G_s                 ▷ eliminate
16         if g ≡ 0 then
17             return False                      ▷ (a, b) is not innovative
18         s ← h(g)
19     G_s ← g                                   ▷ insert g into G
20     for (r, x) ∈ l do
21         y ← y − xY_r                          ▷ replay eliminations l on y using rows of Y
22     Y_s ← y                                   ▷ insert y in Y
                                                 ▷ G is in non-bilateral spread row echelon form.
23     while t^inv_{t(G_s)} ≠ ∅ do
24         (s, r) ← (min{s, ρ}, max{s, ρ}) where ρ = t^inv_{t(G_s)}
25         rear-end eliminate G_s using G_r
26         apply the same row operation on Y
27         t^inv_{t(G_r)} ← r
28     t^inv_{t(G_s)} ← s
                                                 ▷ G is in bilateral spread row echelon form again.
29     return True
```

Listing 6.1.: High-level generic pseudocode of EAGER including conditional row swapping.

decompose the EV of the k^{th} LC, the leading k elements need to be eliminated using row operations requiring $M + M - 1 + \cdots + M - k + 1$ multiplications in total:

$$C(k) = \sum_{i=0}^{k-1} M - i = kM - \frac{k(k-1)}{2}$$

So decomposition of all M EVs requires

$$\sum_{i=0}^{M-1} C(k) = \frac{1}{3}M^3 + O(M^2)$$

(neglecting coefficients of sub-dominant terms) multiplications, the well-known result for Gauss-Jordan elimination that one can find in practically any textbook on numerics [139].

Forward- and Backward Substitution

Visually speaking, every element of L strictly below its diagonal requires one row operation on Y during forward substitution and every element of U on and above its diagonal requires one row operation on Y during backward substitution. Together, this yields M^2 row operations and thus KM^2 multiplications.

Together with the cost of decomposition, this yields

$$\text{cost(LU-based decoding)} = (\tfrac{1}{3}M + K)M^2 + O(M^2)$$

6.4.1 EAGER Decoding

Just like for LU decomposition, we can assume that after k LCs injected, $G_{ij} = 0 \forall j < i$, $G_{ii} \neq 0 \forall i < k$, and $G_{ij} = 0 \forall i \geq k \forall j$. However, due to the property of G being in BSREF we also know that all $t(G_i)$ for $i < k$ are distinct. Eliminating g using G_i requires only $1 + t(G_i) - h(G_i)$ multiplications. Therefore, front-end elimination of the leading k elements of the k^{th} LC's EV requires only $k(M + 1 - k)$ multiplications plus k row operations on Y, since we do not postpone these operations into a later substitution step. This yields a total of $k(M + 1 - k + K)$ multiplications for front-end elimination during injection of the k^{th} LC. Analogously, the following k rear-end elimination steps require $k(M - k)$ multiplications for row operations on G and kK multiplications for row operations on Y. Putting front-end and rear-end elimination together, this yields a cost of

$$C(k) = k(2M + 1 - 2k + 2K)$$

multiplications for the injection of the k^{th} LC. The total cost of injecting M LCs then follows from summation over k:

$$\begin{aligned}\text{cost(EAGER decoding)} &= \sum_{k=0}^{M-1} C(k) \\ &= (\tfrac{1}{3}M + K)M^2 + O(M^2)\end{aligned}$$

Interestingly, this is exactly (when considering only dominant terms) the same result that we computed for LU-decomposition-based decoding. While this is definitely no quantitative benefit over LU decomposition, we still note that with EAGER decoding, we practically get the feature of earliest-possible partial decoding (Proposition 6.6) *for free*, at least asymptotically.

Non-innovative combinations In practice, it is not always possible to design a protocol in a way, such that every node only receives innovative LCs. Therefore, the cost of identifying an LC as non-innovative is still of relevance. In LU decomposition, this cost equals the cost of decomposition: when the decoder already has rank k, it requires $k \cdot (M - \frac{k-1}{2})$ multiplications at worst. In EAGER decoding, this cost does not equal the cost of injection, as neither row operations on Y, nor rear-end elimination takes place. Instead, it equals the cost of EAGER's front-end elimination, namely $k(M+1-k)$ multiplications when the decoder already has rank k. When naïvely assuming a decoder-rank probability distribution for received non-innovative LCs that is flat on $[0, M)_{\mathbb{Z}}$ (which is really naïve, because for $\text{rank} = 0$, an LC cannot even be non-innovative), the expectation value of the computational cost in either case is given by

$$\begin{aligned}\text{cost(LU-based)} &= \frac{1}{M}\sum_{k=0}^{M-1} k \cdot (M - \frac{k-1}{2}) \\ &= \frac{1}{3}M^2 + O(M) \\ \text{cost(EAGER)} &= \frac{1}{M}\sum_{k=0}^{M-1} k \cdot (M+1-k) \\ &= \frac{1}{6}M^2 + O(M)\end{aligned}$$

multiplications to identify a non-innovative LC. In this oversimplified model, EAGER suddenly outperforms LU-based decoding by a factor of two.

6.5 Application of EAGER to Different Codes

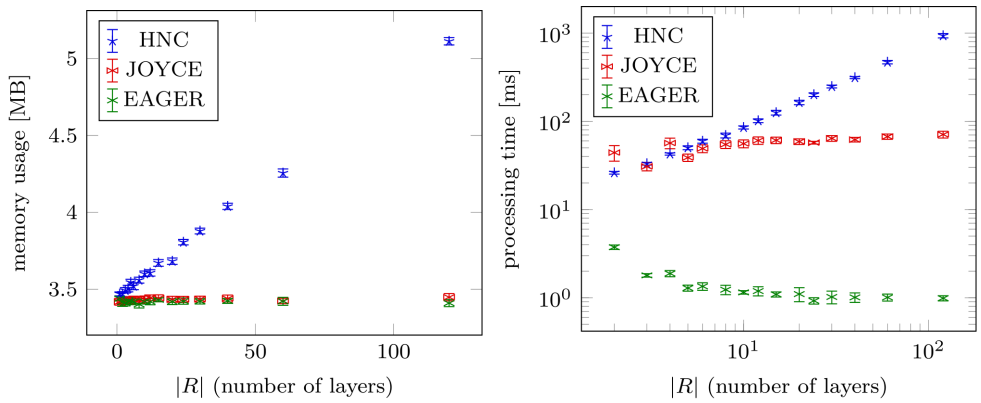
6.5.0 Hierarchical Network Coding

HNC [101] is the ideal example to demonstrate the benefits of EAGER decoding.

- The method introduced so far can be applied directly, without any extension or modification.
- The partial decoding that inherently comes with EAGER is actually the primary goal of HNC.
- We have access to implementation and evaluation source code that was used to evaluate the computational demand of HNC, or JOYCE [88], respectively.
- EAGER decoding is asymptotically more efficient than JOYCE by a factor of M in terms of computation time as well as memory usage.

We first recap the basic idea of JOYCE: starting with empty (i. e., all-zero) $(M \times M)$ matrices L and U, and an identity permutation P, the EV of each injected LC is inserted into the first non-vacant row of U. If the non-vacant rows of U, including the new row, are not sorted by layer, the rows of U are bubble-sorted bottom-up. In order to maintain the invariant $PLU = A$ when swapping adjacent rows of U, it may be required to partially undo the GE operations applied beforehand. This technique is called *counter elimination*. EAGER, using eager forward eliminations and avoiding to store L, does not have this inter-row-dependency: for every row $(G_i|Y_i)$ of $(G|Y)$ holds: $G_iX = Y_i$. Therefore, eager forward substitution allows swapping arbitrary rows in G as long as the corresponding rows in Y are swapped as well, thereby eliminating the need for costly counter elimination. Naïvely, one could think that the well-known strategy of eager forward substitution was sufficient to enable by-layer-ordering with simple row-swapping. But without eager backward substitution, this comes at the cost of performing all forward substitution steps on Y, even if the injected LC is not innovative. For our approach of conditional row swapping, which assures that row swapping never happens in case of non-innovative LCs, eager forward substitution alone is not sufficient, but in addition the concept of the BSREF is required. Thus, while eager forward substitutions and direct swapping of non-adjacent rows could be used as an alternative approach to LU decomposition and counter elimination, only EAGER backward substitution assures that solving the system for a given right-hand side Y can be performed in less than or equal to M^2 elementary row operations on Y.

In order to compare EAGER to JOYCE in terms of processing time and memory usage, we take a look at the model that is defined by JOYCE's authors [88]: it is assumed that a



(a) Memory usage.

(b) Processing time.

Fig. 6.2.: Comparison of HNC, JOYCE, and EAGER in terms of (a) memory usage and (b) processing time for decoding a generation of size $M = 120$ depending on the numbers $|R|$ of layers. Error bars depict 95% confidence intervals (CIs).

generation of size M is divided into $|R|$ fine-grained prioritization layers, meaning that $|R| \in \Theta(M)$, or equivalently that $m := \frac{M}{|R|}$ is constant. Further it is assumed that a decoder receives LCs mostly in order, i. e., with a maximum deviation of C layers, where $C \in \mathbb{N}$ is independent of M. We call this the in-order assumption (IOA). The opposite, i. e., not assuming any special ordering of injected LCs, would translate to $C \in \Theta(M/m)$.

6.5.1 Empirical Comparative Performance Evaluation

To give JOYCE a fair chance, we start by approximately reproducing the performance evaluation carried out in [88], only adding data series for EAGER. In Fig. 6.2 we show measurements of memory usage and processing time of HNC, JOYCE, and EAGER for decoding a generation of size $M = 120$ with varying number $|R|$ of equally-large layers. We also adopt the assumption of the sender's imperfect knowledge about the receiver's decoding state: if the decoder has already decoded the first k layers successfully, instead of injecting combinations of the best-fitting generation k, we chose a layer uniform at random from $\{k-C, \dots, k+C\} \cap [0, |R|)$. The data clearly confirms that EAGER's computational demand for decoding is essentially independent of layering, just as it is the case for JOYCE. As it is not possible in case of EAGER, to measure the time for decomposition and substitution separately, the measured processing times incorporate the complete solution of the system of linear equations for each decoding algorithm. However, in order to focus on solving the encoding matrix instead of the coded data part, we chose $K = 1$ for all timing experiments.

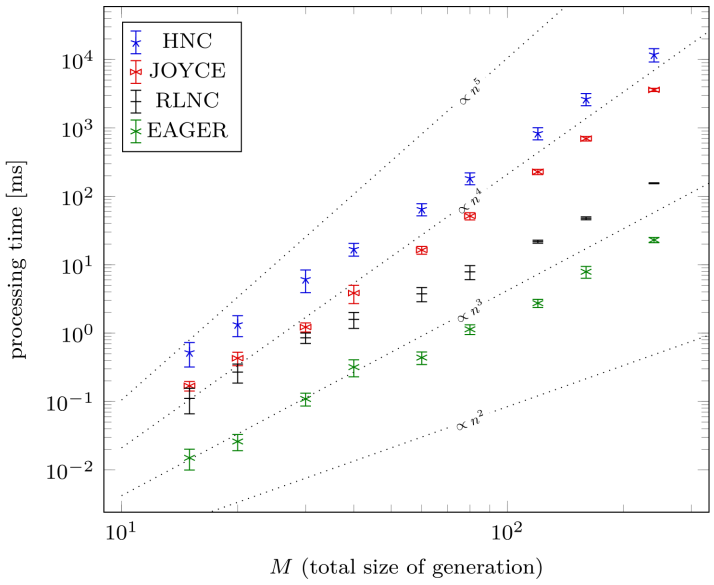
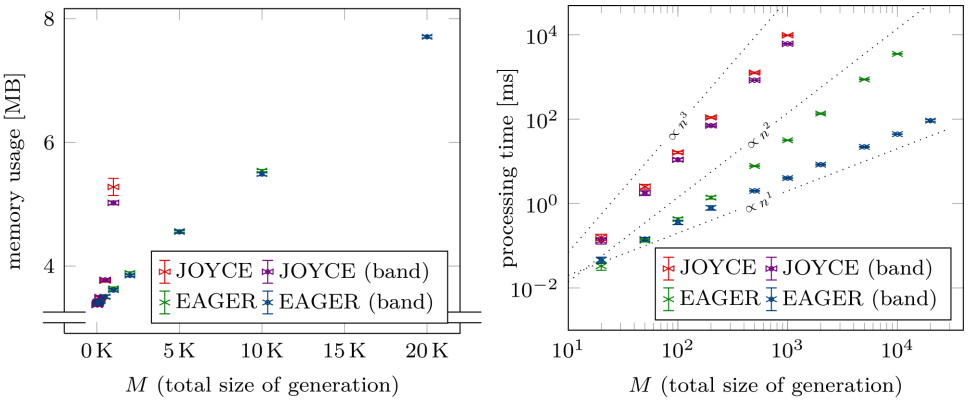


Fig. 6.3.: Processing times of different decoding algorithms for layered prioritized RLNC, given an ignorant sender that selects layers uniform at random. Dotted straight lines are drawn only to give orientation regarding slopes. Error bars depict 95% CIs.

In addition, we also repeated the measurement of processing time under the worst-case assumption that the sender has no knowledge about the decoder's state at all: in Fig. 6.3 we considered a sender that generates packets by choosing the layer uniform at random from all possible layers and measured the processing time for complete decoding depending on the generations size M for fixed $m = 1$. In addition to HNC, JOYCE, and EAGER, we also take a vanilla not-partially-decoding LU decomposition-based decoder into consideration, denominated "RLNC" in the plot. The data confirms that both HNC and JOYCE have a processing time of $O(M^4)$, whereas vanilla RLNC and EAGER require $O(M^3)$. In addition, we learn that EAGER performs roughly a factor of ten faster than vanilla LU-decomposition-based RLNC, thereby demonstrating a rather mundane benefit over LU decomposition that, admittedly, might be an artifact of the specific implementations.

The strength of EAGER, when applied to prioritized layered RLNC, becomes much clearer when considering resource utilization depending on generation size M for a fixed layer size m (as opposed to Fig. 6.2, where M is fixed and m is varied). For $m = 10$ and layer-imperfection $C = 2$ we measured memory usage and processing time of EAGER and JOYCE for a wide range of values of M; results are given in Fig. 6.4. In addition to the *bulk layered* approach, where an LC corresponding to the k^{th} layer has non-zero coefficients in $[0, (k+1)m)_{\mathbb{Z}}$ we also measured performance in the case of a *band layered* code where encoding vectors of a layer have the same zero-tail properties, but at most



(a) Memory usage.

(b) Processing time.

Fig. 6.4.: Comparison of JOYCE and EAGER in terms of (a) memory usage and (b) processing time for decoding. Generation size M is varied for constant layer size m. Both algorithms are evaluated for bulk layered coding as well as for band layered coding. Dotted straight lines are drawn only to give orientation regarding slopes. Each data point has been measured with 10 differently seeded runs. Error bars depict 95% CIs. There are no data points for JOYCE at $M > 1000$ because each measurement run was performed with a timeout of 10 s.

$\ell \in \mathbb{N}_+$ consecutive coefficients are non-zero. The terms "band"/"bulk" layered are our own creation and do not stem from the related work.

$$\text{bulk LC of layer } k\text{:} \qquad a_{ij} = 0 \ \forall i \notin [0, (k+1)m)_{\mathbb{Z}}$$
$$\text{band LC of layer } k\text{:} \qquad a_{ij} = 0 \ \forall i \notin [\max\{0, (k+1)m - \ell\}, (k+1)m)_{\mathbb{Z}}$$

An example of sparsity patterns of EVs and decoder state matrices for bulk layered and band layered codes for $m = 3$ and $\ell = 6$ is given in Fig. 6.5.

We find that for either encoding strategy EAGER has memory usage linear in the size of the generation, while JOYCE's memory usage is super-linear. When considering the sparsity patterns of matrices L and U of JOYCE for layered band-coded LCs as depicted in Fig. 6.5, there is reason to presume that JOYCE's memory usage could also be made linear by changing its implementation appropriately to use sparser data structured to store the rows of L and U. The comparison of processing times however shows that EAGER can achieve linear time for band-coded LCs and quadratic time for bulk-coded LCs. We note that these processing times are asymptotically optimal because they are proportional to the number of non-zero random coefficients, each of which needs to be processed for decoding. We conclude that EAGER is a promising concept as its resource utilization is not only asymptotically optimal for large M but also smaller than its contenders for small generation sizes that would be expected in practical applications.

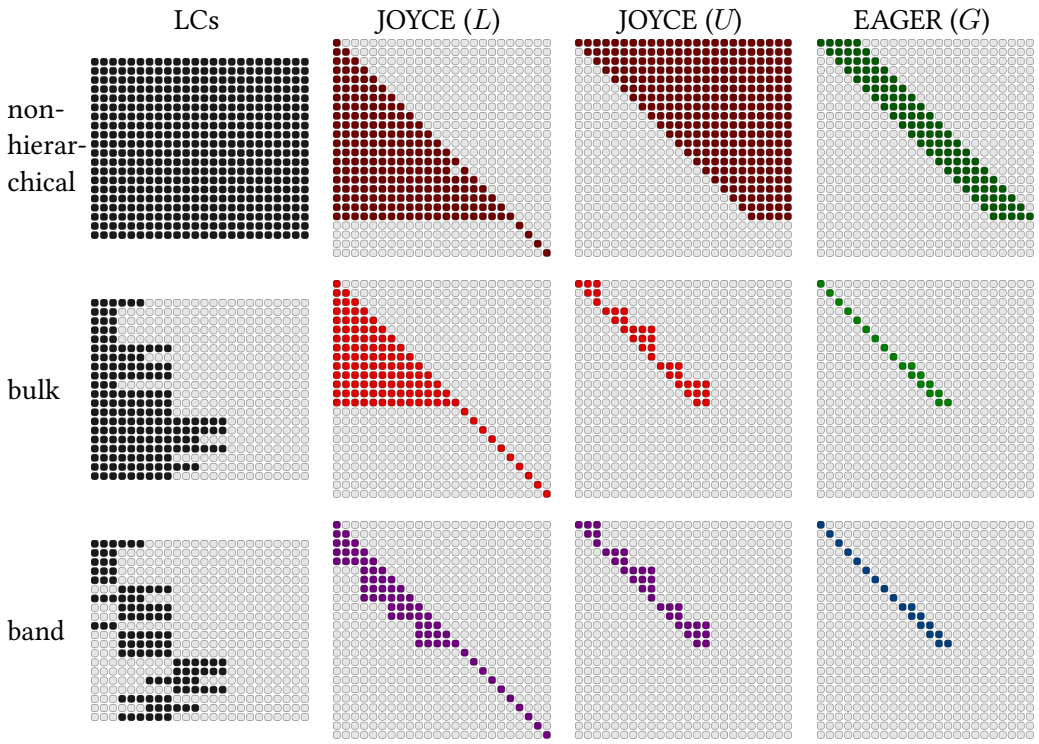


Fig. 6.5.: Example snapshots of LCs and decoder state sparsity patterns for JOYCE's L and U matrices as well as EAGER's G matrix for $M = 24$, $m = 3$, $C = 2$ bulk and band codes, each after the injection of 20 LCs.

6.5.2 Treatment of Wrapped Interval-Sparse Codes

As we have already seen, EAGER is well-suited to decoding and recoding non-wrapped interval-sparse random linear codes. For the use case of bulk broadcast using fountain/network hybrid random linear codes that we treat in Chapter 7, however, we use end-wrapping chunked overlapped random linear codes because of their rotation symmetry in the index space.

We discuss two strategies for decoding codes with end-wrapping interval-sparse EVs. Consider for example, for an $M = 8$, $\ell = 4$ interval-sparse code, an encoding vector $\mathbf{a}_i = (1, 1, 0, 0, 0, 0, 1, 1)$. We can straight-forwardly inject this LC into an EAGER decoder, ignoring its sparsity and treating it as bulk encoding vector with $h(\mathbf{a}_i) = 0$ and $t(\mathbf{a}_i) = 7 = M - 1$. If decoding was the decoder's only purpose, this was a perfectly valid strategy. Even though EAGER performs particularly efficiently when applied on interval-sparse encoding vectors, there is no requirement that EVs have any sparsity pattern at all.

Recoding, on the other side, is significantly hampered by this strategy. When not-end-wrapping LCs are injected into an EAGER decoder, their sparsity and therefore their recodability is only increased, because front-end and rear-end elimination never enlarges the non-zeros' convex hull. (By *convex hull* we here mean the index-space-interval covering all non-zero elements of a vector or of a matrix row.) The sparsity of an end-wrapping LC on the other side lies in a stride of zero coefficients in the middle of two non-zero ranges. This kind of sparsity is by no means conserved by EAGER.

To enable both recoding and decoding for end-wrapping interval-sparse codes, we use a simple trick that allows us to insert end-wrapping EVs without tearing them apart: Consider a code with fixed M and ℓ whose sparsity classes are all representable as

$$C = [h, h + \ell)_{\mathbb{Z}} \qquad \text{with } h \leq M - \ell \tag{6.4}$$

or

$$C = [h, M)_{\mathbb{Z}} \cup [0, h + \ell - M)_{\mathbb{Z}} \qquad \text{with } h > M - \ell \tag{6.5}$$

The end-wrapping type of classes (6.5) can be re-written as simple (non-wrapping) integer intervals

$$C = [h, h + \ell)_{\mathbb{Z}} \qquad \text{with } h > M - \ell \tag{6.6}$$

if we logically extend the matrix X in an M-periodic fashion to shape $M' \times K$ where $M' = \max\{\, h + \ell \mid [h, h+l)_{\mathbb{Z}} \in \mathcal{C}_{(6.6)} \,\}$:

$$X_k := X_{k-M} \,\forall\, k \geq M \tag{6.7}$$

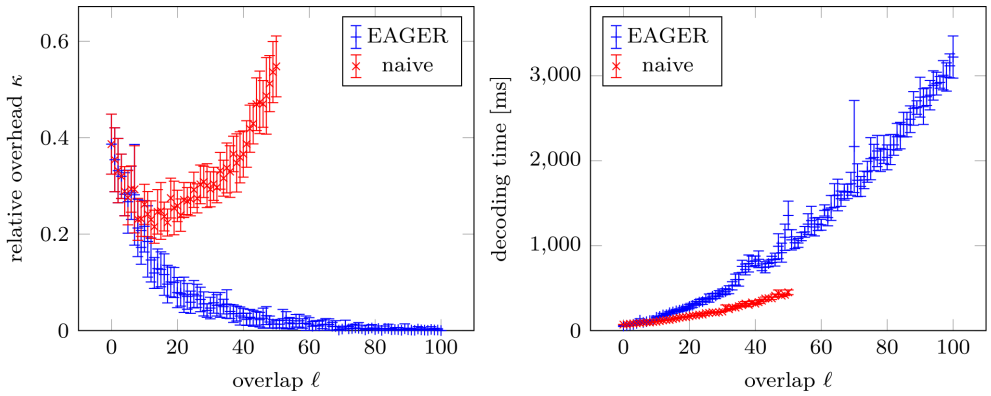
When we rewrite every end-wrapping EV in this way, they have non-zero coefficients at indices greater than or equal to M. In order to inject these EVs into the decoder matrix G, it needs to be enlarged to $M' \times M'$ as well as Y which now is of size $M' \times K$. At a first glance, it seems that we have bought the ability to efficiently recode packets at the cost of needing M' instead of M innovative packets for G to become diagonal and thus to being able to decoding the whole transmission. This is caused by the fact that we use (6.7) for encoding, but have not used it for decoding so far. Therefore, we rewrite (6.7) in row-matrix form as

$$\begin{aligned} X_{k-M,j} - X_{k,j} &= 0 && \forall\, k \in [M, M')_{\mathbb{Z}} \forall j \in [0, K)_{\mathbb{Z}} \\ \underbrace{(\overbrace{0, \dots, 0}^{k-M}, 1, \overbrace{0, \dots, 0}^{M-1}, -1, \overbrace{0, \dots, 0}^{M'-k-1})}_{\text{encoding vector}} \cdot X &= \underbrace{(\overbrace{0, \dots, 0}^{K})}_{\text{information vector}} && \leftarrow \text{ matrix equation} \end{aligned} \tag{6.8}$$

and simply inject these $M' - M$ LCs that encode the redundancy in X' into the decoder during initialization, i. e., before the first real LC is injected. Note that by -1 we mean the additive inverse of 1 which is equal to 1 for finite fields whose order are integer powers of 2. Since the decoder state matrix is hereby initialized with rank $M' - M$, only M additional innovative LCs are required to achieve full rank and thereby decodability.

End-Wrapping Overlapped Chunked Codes: EAGER vs. Separate Decoders

As stated earlier, EAGER decoding lends itself to decode overlapped chunked random linear codes [103]. When dividing M source symbols into $N = M/(m - \ell)$ chunks of size m such that every two contiguous chunks overlap by ℓ symbols in an end-wrapping fashion, decoding can be performed in two fundamentally different ways: one can either use one large joint decoder of size M (or rather $M' = M + \ell$ when using EAGER), or one could instantiate N separate decoders of size m each, one per chunk, and inject each received combination into the corresponding decoder. In the case of separate decoders, every time one of the per-chunk decoders reaches full rank m, the chunk can be decoded and ℓ trivially encoded LCs (whose EV is a standard basis vector) can be injected into its predecessor and successor chunk each.



(a) Code overhead.

(b) Decoding time.

Fig. 6.6.: Comparison of decoding strategies for end-wrapping overlapped chunked RLNC. Per-chunk decoding and joint decoding using EAGER is compared in terms of (a) code overhead and (b) compute time for decoding. Error bars depict standard deviation.

The joint decoding approach has two advantages over per-chunk decoding: first, it can decode sets of LCs that are not decodable with separate decoders because the latter require that there is at least one chunk of which m linearly independent LCs have been received. Second, joint decoding using EAGER improves the *recodability*, because LCs that belong to one chunk implicitly contribute to neighboring chunks as soon as more than $m - \ell$ linearly independent LCs have been injected. Using separate decoders, this inter-class recoding happens only when chunks reach full rank.

To quantify the advantage of joint decoding using EAGER over separate per-chunk decoders empirically, we consider a single-hop scenario consisting of one source node emitting LCs with $m-\ell = 50, q = 100$ and a destination node that receives packets and seeks to decode the message. In each run and for each decoding approach we measure the number n of random LCs that is needed to decode the message entirely (which is termed *capacity* by some authors [103]). For each decoder and each value of ℓ we performed 10 independently seeded runs. The relative amount of excess packages $\kappa = \frac{n}{M} - 1$ is also known as *code overhead*.

In Fig. 6.6a we see that for both decoding approaches the overhead benefits from overlapped chunks if the overlap is small ($\ell \ll m$). However, the overhead of per-chunk decoding raises again as $\ell \to 2(m - \ell)$ while the overhead of joint decoding seems to decrease monotonically and to approach 0 asymptotically. The latter is consistent with analytic results found in [104].

It turns out, however, that the reduction of overhead comes at an increased computational demand, as can be seen in Fig. 6.6b.

rank comparison	implication on vector space	innovativity of random LC
$\operatorname{rank} G^A > \operatorname{rank} G^B$	$\operatorname{span} G^A \not\subseteq \operatorname{span} G^B$	$A \to B$ is innovative
$\operatorname{rank} G^A < \operatorname{rank} G^B$	$\operatorname{span} G^A \not\supseteq \operatorname{span} G^B$	$B \to A$ is innovative
$\operatorname{rank} G^A = \operatorname{rank} G^B$	??	??

Tab. 6.1.: Deduction of innovativity of randomly recoded packets from decoder state ranks only.

6.6 Normalization of Decoder State

6.6.0 Motivation

In a scenario with nodes using RLNC to transmit packets over multiple hops, a forwarding node needs to decide whether its decoder contains information that is innovative for one or more of its neighboring nodes. Let us consider two nodes A and B that are immediate neighbors and use chunked RLNC to distribute data. Without loss of generality looking at the perspective of node A, it has to decide for each chunk index i whether an LC that is randomly recoded from its own decoder state is likely to be innovative for node B. To avoid visual clutter, we will not write out the index i in this subsection, but simply write G^A and G^B to refer to the nodes' decoder state matrices G restricted to the rows whose non-zero elements lie entirely within chunk i's sparsity class. In other words: node A has to judge whether the vector space spanned by its own decoder state is a subspace of the space spanned by B's decoder (in which case an LC of A cannot be innovative for B). A plain representation of the entire coefficient matrix that represents the decoder state is typically far too large to be shared between different nodes: when assuming packets of approximately 256 B and $\mathbb{F}_{256}$, already the decoder state corresponding to a small chunk size of 16 would fill an entire packet. Therefore, we can assume that including one or more chunks' decoder coefficient matrix in packet headers implies at least a significant overhead and is unfeasible for larger chunk sizes.

A more lightweight heuristic to decide whether A's packets are innovative for B relies on ranks of the decoders' coefficient matrix. The rank of the coefficient matrix of a chunk of size m is an integer in $[0, m]$ and can thus be encoded in $\lceil \log_2(m+1) \rceil$ bits (compared to m^2 bytes to encode the plain coefficient matrix). If the RLNC protocol distributes per-chunk decoder ranks as meta information such that neighboring nodes are aware of each other's chunk ranks, the Table 6.1 describes the implied innovativeness of packets.

Unless $\operatorname{rank} G^A = \operatorname{rank} G^B$, one of the nodes can be certain that its combinations will be innovative for the other node. Therefore, a conservative approach would let nodes transmit combinations of a chunk if and only if its decoder rank is strictly greater than the receiver's rank. Naturally, when considering only these two nodes, this tends toward

a situation where both nodes have the same rank and no node will transmit anymore. In static topologies with a single source node that has full-rank information of all chunks, this is not much of a problem: intermediate nodes will typically have smaller ranks the further downstream from the source they are located. The source (which has full rank and therefore higher rank than any not-finished node) can permanently pump LCs into its one-hop neighborhood. Therefore, all nodes in the network can keep emitting LCs based on the conservative rank heuristic and the only steady state[7] that can be reached is the state where all nodes in the network have full rank.

In Chapter 7 we study topologies where the source node distributes LCs to different intermediate nodes before it goes out of communication range for a while. This can lead to a situation where the vector space spanned by the union of all intermediate nodes' decoders has full rank, even though no decoder by itself has full rank. Using the conservative rank-based heuristic for recoding could (and sometimes does) then reach a steady state, where all intermediate nodes have reached the same (non-full) rank, even though their spanned vector spaces differ. In other words: in some situations, the conservative heuristic can tell every node not to transmit, even though some or even every node has innovative LCs for its neighbors.

One possibility to overcome this steady state of unequal decoder subspaces is to find a representation of the decoder state that uniquely depends on the spanned vector space and is independent of the LCs that were injected into the decoder. We could then feed this unique state into a hash function h and exchange the resulting hash values between nodes. If nodes A and B find that they not only have decoder states of equal rank but also of equal hash value, both know (up to hash collisions) that their decoders represent equal vector spaces and therefore neither of them can transmit an innovative LC to the other. If, on the other hand, the ranks are equal, but the hash values are not, it is clear that neither A's vector space is a subspace of B's, nor vice versa. For decoders of unequal rank, the hash value does not contribute any additional information. Table 6.1 summarizes the implication of different combinations of rank- and hash-relation on vector space relation and thereby innovativity of transmitted LCs.

Thus, when we include hashes of all chunks' decoder states into the meta-information shared between neighbors, the only steady state (aside from hash collisions) of the resulting hash-extended conservative heuristic is the state of all nodes decoder states being equal.

[7] By *steady state* we mean, in this context, a state where all nodes have stopped to transmit LCs and therefore the data dissemination has stalled.

rank comparison	hash values	implication on vector space	innovativity of random LC
$\operatorname{rank} G^A > \operatorname{rank} G^B$	irrelevant	$\operatorname{span} G^A \not\subseteq \operatorname{span} G^B$	$A \to B$ is innovative
$\operatorname{rank} G^A < \operatorname{rank} G^B$	irrelevant	$\operatorname{span} G^A \not\supseteq \operatorname{span} G^B$	$B \to A$ is innovative
$\operatorname{rank} G^A = \operatorname{rank} G^B$	$h(G^A) \neq h(G^B)$	$\operatorname{span} G^A \not\subseteq \operatorname{span} G^B$ $\operatorname{span} G^A \not\supseteq \operatorname{span} G^B$	$A \leftrightarrow B$ both innovative
$\operatorname{rank} G^A = \operatorname{rank} G^B$	$h(G^A) = h(G^B)$	$A = B$	none (unless hash collision)

Tab. 6.2.: Deduction of innovativity of randomly recoded packets from decoder state ranks and decoder state hash values. See also Table 6.1.

6.6.1 A Unique Representation of the EAGER Decoder's State

All we need in order to achieve a unique representation of the decoder state is a set of properties of the coefficient matrix such that given one coefficient matrix G that satisfies all properties, any non-trivial row operation on G breaks at least one condition.

The first condition that we use is that G is in BSREF, which is already maintained by EAGER by default. Fortunately, BSREF does not leave many ambiguities for the coefficient matrix. The first ambiguity lies in constant per-row pre-factors: multiplying G with a non-singular diagonal matrix changes the coefficients in G but leaves it in BSREF. Therefore, we require as second uniqueness condition that the first non-zero element in every row of G shall be 1 (which is an arbitrary but obvious choice). The only remaining ambiguity is that we can add a multiple of one row r to another row s where $s < r$, if $t(G_r) < t(G_s)$:

$$G'_{si} = G_{si} + aG_{ri} \; \forall i \tag{6.9}$$

Remember that for each non-vacant row s, s is the index of the first non-zero element. Thus, for $r < s$, $G_{sr} = 0$ and the row operation (6.9) for $a \neq 0$ would result in $G'_{sr} \neq 0$, breaking BSREF. Obviously, the same argument holds for the indices $t(G_s)$, $t(G_r)$ of the last non-zero element of each row, which is unique in BSREF. If $t(G_r) > t(G_s)$ then $G'_{s,t(G_r)} = aG_{r,t(G_r)} \neq 0$ would be the last non-zero element of row G'_s, breaking BSREF. Therefore, row operations of type (6.9) only maintain the BSREF condition if

$$s < r \wedge t(G_s) > t(G_r), \tag{6.10}$$

i. e., if the convex hull of non-zero element indices of G_r lies entirely within the corresponding convex hull of non-zero element indices of G_s. To remove the ambiguity implied by this kind of row operation, we require that for every pair (s, r) with $s < r$ and $t(G_s) > t(G_r)$: $G_{sr} = 0$. If this condition is not satisfied after injection, it can be established by the row operation (6.9) with $a = -G_{sr}/G_{rr}$. Since this elimination produces a 0 somewhere between G_{ss} and $G_{st(G_s)}$, we term this a *middle elimination*.

Fortunately, this rarely happens when using sparsity classes of equal length, or at least when using an overlapped chunked linear code, as we investigated empirically.

We call a coefficient matrix G satisfying these three conditions to be in *reduced* BSREF in analogy to the (equivalently unique) *reduced row echelon form* that is well known from linear algebra. Once the decoder is adapted to bring G into reduced BSREF after each injection, we can easily compute hash values of either the whole matrix G or of sub-matrices consisting of the rows that correspond to a certain sparsity class. We use a chunked RLNC approach where each transmitted LC fits into the sparsity class of a chunk in Chapter 7. Therefore, we compute submatrix hash values corresponding to the chunks.

We note here that if only per-chunk hashes are ever used, it is sufficient to restrict conditions (2) and (3) to rows that fall into any chunk's sparsity class. Especially the de-wrapping rows (6.8) receive many non-zero elements by elimination and often break these conditions. But as long as their non-zero convex hull is longer than any chunk's sparsity class, they do not directly take part in recoding, and they are therefore excluded from hashing, and thus also excluded from middle elimination in our implementation.

To hash a submatrix of G consisting of k non-vanishing rows in the corresponding sparsity class, we push the concatenated byte sequence

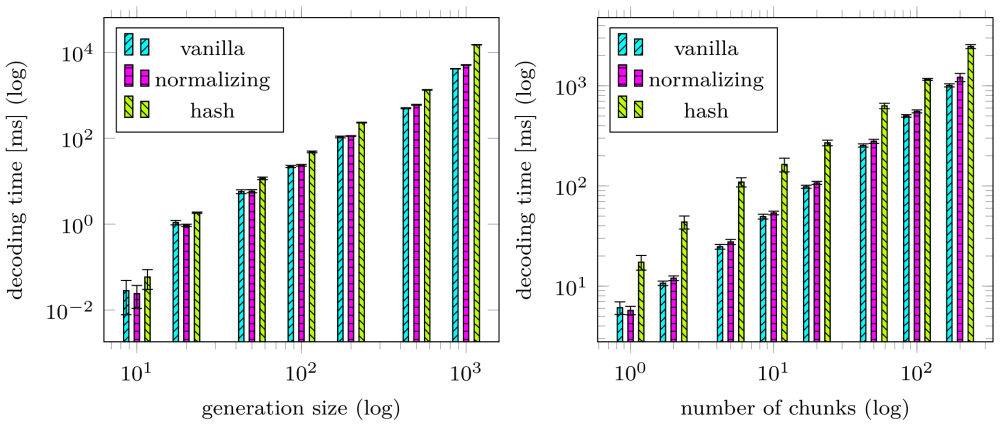
$$(s_0, t_0, G_{s_0 s_0}, G_{s_0 s_0+1}, \dots, G_{s_0 t_0}, s_1, t_1, G_{s_1 s_1}, \dots, G_{s_{k-1} t_{k-1}})$$

into a hash function capable of digesting byte streams of arbitrary length. In our implementation we use `SpookyHashV2` [140] for this purpose.

6.6.2 The Computational Cost of Decoder State Uniqueness

Adding rear-end elimination to the incremental Gaussian elimination does not incur any additional cost, because (a) the additional row operations during one injection are later repaid by reducing the cost of subsequent front-end eliminations and (b) the resultant shortening of rows of G is never cancelled by subsequent elimination steps. See Section 6.4 for a full discussion of this topic.

This does not hold for additional middle elimination operations. The zero elements in the middle of G's rows are likely to be cancelled by subsequent front-end or rear-end elimination steps and even if they did not, they would not shorten the ranges of loops during elimination and therefore would not save any operations. The search effort to find all ordered pairs of rows satisfying (6.10) after every successful injection incurs a secondary



(a) Bulk code.

(b) Overlapping chunked code.

Fig. 6.7.: Compute time for decoding using vanilla EAGER, unique-decoder-state EAGER, and unique-decoder-state EAGER computation of all chunk hashes after each injection. (a) bulk code with varying generation size and (b) end-wrapping overlapping chunked code with chunk size 50, overlap 10 and varying number of chunks. Error bars depict standard deviation.

computational cost of the unique representation. Additional memory is, however, not required for middle elimination.

To quantify the cost of achieving a unique representation of decoder state in practice, we measured the total decoding time required for bulk code as well as end-wrapping overlapping chunked code. We compare decoding time without reduced BSREF, with reduced BSREF, and with reduced BSREF plus computing the hash value of every chunk (that changed) after every injection. The results given in Fig. 6.7 indicate that establishing the unique representation of decoder state incurs a constant factor for 0.1–0.2 of additional computation time. The actual computation of hash values increases the total computation time by a factor of 2–3. However, a strong dependence on the total system size or chunk overlap size cannot be observed in this data.

6.7 Conclusion

We introduced EAGER, a novel algorithm to solve systems of linear equations over finite fields to be used in RLNC. We have shown both analytically and empirically that EAGER can outperform RLNC decoders that are based on LU decomposition. EAGER can seamlessly handle the joint decoding of overlapped chunked random linear codes, thereby decreasing decoder-induced overhead, at least in end-to-end fountain code scenarios. Finally and most importantly, EAGER is, to the best of our knowledge, the first RLNC decoder that is capable of maintaining its decoder in a reduced state that depends only on the vector space

spanned by the injected LCs. As we demonstrate in Chapter 7, this feature can be used to simplify RLNC protocol design in the context of Over-the-Air Programming (OTAP) in low Earth orbit (LEO) satellite formations.

7 Over-the-Air Programming of Satellite Formations Using Random Linear Network Coding

> *Alles Ständische und Stehende verdampft, alles Heilige wird entweiht, und die Menschen sind endlich gezwungen, ihre Lebensstellung, ihre gegenseitigen Beziehungen mit nüchternen Augen anzusehen.*
>
> — **Karl Marx und Friedrich Engels**
> from: Manifest der Kommunistischen Partei

TL;DR *In satellite OTAP based on RLNC, a decoder state that uniquely depends on the represented subspace helps to improve communication efficiency.*

7.0 Introduction

Nano satellite formations will be implemented that provide cooperative attitude control capabilities for simultaneous target observations, e. g., in the missions TOM [105] and CloudCT [106]. In the CloudCT project a formation of ten satellites will be implemented by the Zentrum für Telematik e.V. (ZfT) in Würzburg, Germany. 3D information of clouds will be captured by a satellite formation flying in a dense orbit configuration, as displayed in Figure 7.0. While the satellite mission's primary goal is the acquisition of data in orbit which is then forwarded to the ground segment, satellites require frequent reprogramming in terms of either regular command and control or in terms of firmware updates that need to be deployed to each craft in the formation. Even though quite similar to OTAP in wireless sensor networks (WSNs) at a first glance, the time-varying topologies and channel conditions are significantly different from terrestrial sensor networks. The dynamical topology of a whole system comprising one ground station (GS) and a number of satellites in dense orbit configuration has three outstanding characteristics: First, the

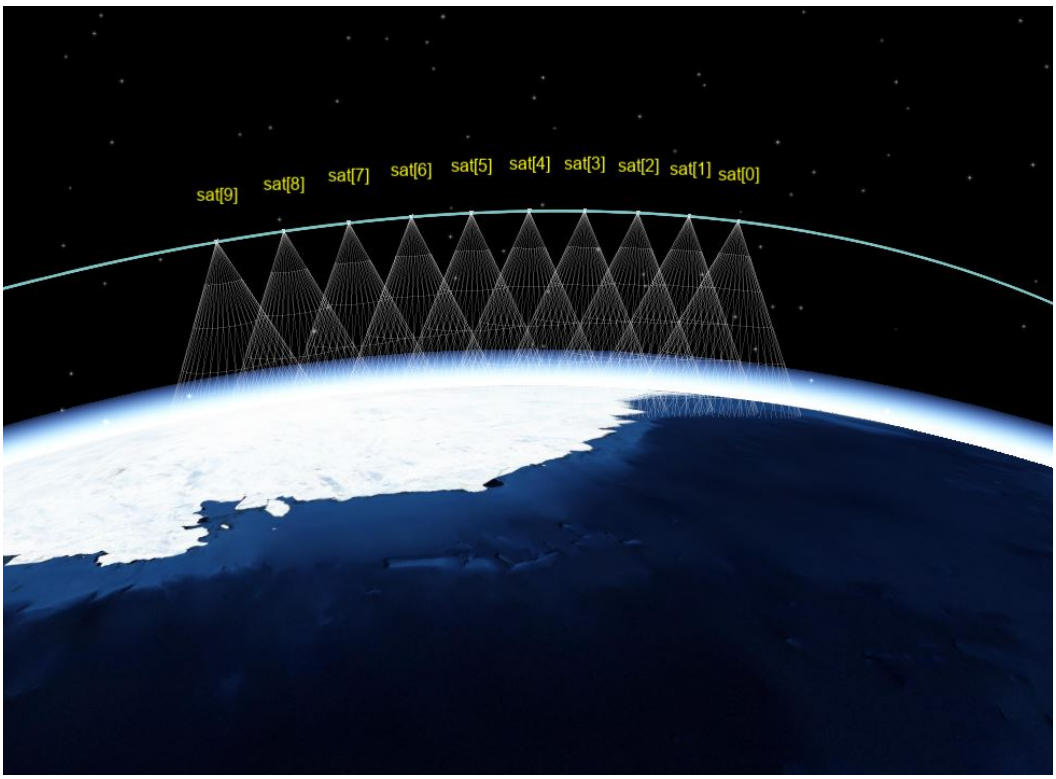


Fig. 7.0.: The CloudCT string-of-pearls orbit configuration.

satellites' geometrical constellation is highly regular in space and constant over time, at least when considering a string-of-pearls configuration as planned for the CloudCT mission. Second, the satellites' effective topology in terms of connectivity and packet loss is neither constant nor predictable: it may be largely restricted by third party terrestrial interferers like surveillance radar [107]. Connectivity may therefore significantly change as the satellites move with respect to the Earth's surface. Third, the satellites' connectivity to the GS is somewhat special: for most orbital configurations with homogeneous altitude and inclination, the long-term average connectivity of each satellite to the GS is identical for all satellites of the formation. Thus, none of the nodes is naturally preferential to be the GS's first hop for broadcast data dissemination. Furthermore, the GS's aggregate communication windows to all satellites in the formation constitutes only a small fraction of time. Therefore, in-orbit dissemination via inter-satellite links may significantly reduce delay when the same transmission payload is to be deployed to all satellites.

The task of broadcasting one bulk payload from the GS to all satellites in the formation naturally lends itself to being solved with either a fountain code (also known as *rateless code*) or with RLNC. On the one hand, adjacent satellites are close enough that most packets transmitted by the GS can be received by more than one satellite. The formation's total extent, on the other hand, is so large that no packet transmitted by the GS can be

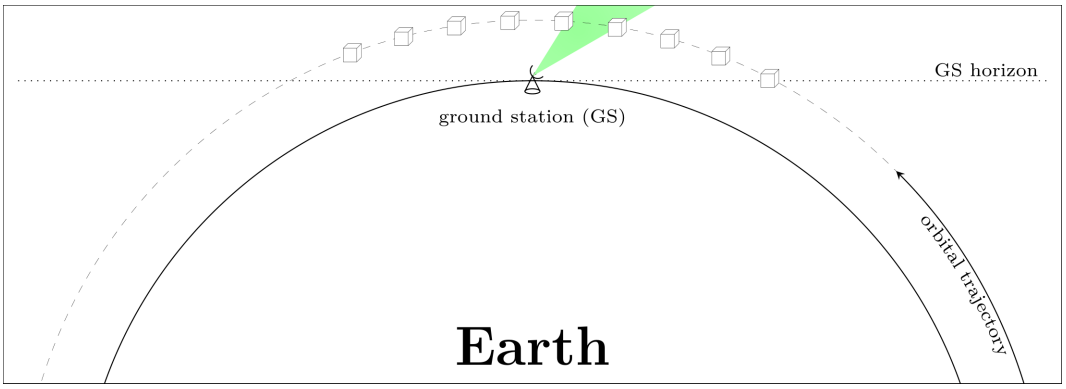
received by all satellites at once (see Fig. 7.1). This implies that, considering any pair of adjacent satellites, their sets of received packets have a large intersection, but neither is a subset of the other. In such a situation, the total number of transmissions required to complete the bulk broadcast transmission can be significantly reduced by using a fountain code [6]. In simple terms, a fountain code is an erasure code that defines a practically unbounded number of code words of size K such that any random $\lceil \mathcal{S}/K \rceil$-element subset of code words is sufficient to decode the source of size $\mathcal{S}$ with high probability.

While using a fountain code has significant advantages over naïvely chopping the bulk payload into uncoded segments, it still requires that every satellite receives the entire coded transmission directly from the GS. Using inter-satellite transmissions, the GS-to-satellite communication demand can be further reduced: as soon as the set of all packets received by any satellite of the formation encodes the entire payload, satellites could exchange data among each other, let it be by forwarding of received packets or by recoding (RLNC), until each satellite can decode the transmission.

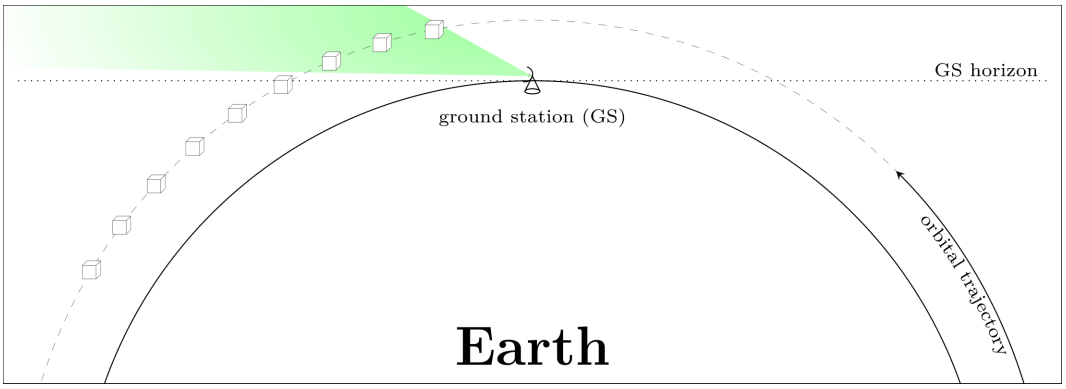
In order to minimize the total transmission delay or the GS-to-satellite uplink channel resources, this use of inter-satellite communication is especially appealing when considering that a quasi-static inter-satellite topology offers significantly higher long-run average channel capacity than the rare and short GS-to-satellite contact periods that are typical for LEO satellite formations. In this chapter we use the term *contact period* to refer to a (maximal) interval of time when there is at least one of the formation's satellite above the GS's horizon.

In this chapter we introduce a simple chunked-RLNC-based toy protocol for OTAP. Since every transmission in chunked RLNC belongs to one chunk, the protocol needs to decide for any given transmission opportunity whether to transmit at all as well as from which chunk to create a code word. In a simulation-based empirical evaluation, we use the toy protocol to compare different chunk selection strategies with respect to use-case-motivated performance metrics. In particular, we want to answer whether the availability of RLNC decoder states that depend uniquely on the encoded linear subspace leads to measurable performance gains. In addition, we also compare the toy protocol to a simple fountain-code-based single-hop broadcast strategy to highlight the benefits of RLNC-based inter-satellite data dissemination over a single-hop fountain code approach empirically.

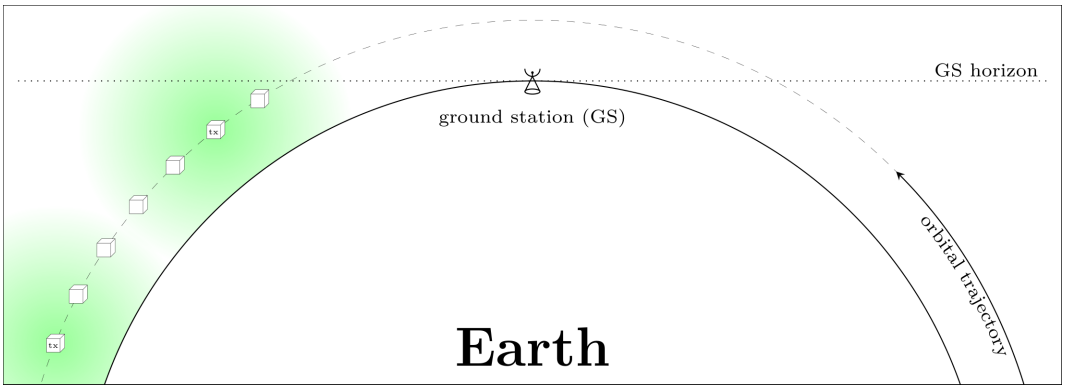
The rest of this chapter is structured as follows: an overview of related work, especially explaining why we abstain from including existing OTAP protocols in our evaluation, is given in Section 7.1. In Section 7.2 we introduce the toy protocol, a minimalist chunked-RLNC-based OTAP protocol for string-of-pearls satellite formations. In a simulation-based empirical evaluation that we present in Section 7.3 we use this toy protocol to compare



(a) All satellites above GS horizon.



(b) Some satellites above GS horizon.



(c) No satellite above GS horizon.

Fig. 7.1.: Schematic overview of a network consisting of one satellite ground station and a string-of-pearls LEO satellite formation. Even when all satellites are above the horizon (a), they may not fit simultaneously in the GS antenna's main lobe. As typical for LEO formations in dense orbital configuration, most of the time ($> 90\%$) no satellite at all is above the horizon (c). (Dawing is not to scale.)

different chunk selection strategies against each other. Concluding remarks follow in Section 7.4.

7.1 Related Work

OTAP (also abbreviated "OAP" by some authors) in WSNs has attracted a lot of attention in the research community during the last two decades. One of the earliest and most cited OTAP protocols is Deluge [108], implemented in TinyOS. Many authors proposing different approaches for OTAP compare their work to the Deluge protocol which despite its popularity has been shown to suffer severe scalability issues, e. g., the "NACK implosion problem"[100].

In order to reduce control message overhead created by ACK and NACK messages and to improve the benefits of multiple nodes receiving a message, several multi-hop OTAP protocols based on rateless codes and/or network coding have been proposed. In the following we discuss the applicability of some of these protocols with respect to the satellite OTAP problem. We note here that since "packet" is already heavily used in RLNC for what we call "linear combinations," and since "packet" is, in a different context, a common term for network-layer objects, i. e., data objects that are transmitted over multiple hops, we rather use the term "message"[108] for the data that is contained as payload in a Medium Access Control (MAC) frame. Unfortunately, in OTAP it not uncommon [100] to use the term "packet" for both, "message" and "uncoded row". However, in this chapter we still sometimes use the term "packet" as a synonym for "message" in common terms like "packet loss."

Most of the following protocols divide the bulk payload (e. g., a firmware image) into a number of smaller units called *pages*, each of which is then further divided into several messages or code words. These pages are what we called *chunks* in Chapter 6. All the equivalent terms "chunk" [98], "generation" [102], "class" [97], and "aperture" [103] are common in RLNC literature and refer to the subsets of all uncoded symbols from which linear combinations are constructed. In OTAP, the term "page" [108, 100] is more common, but some authors also use "batch" [99] for the very same thing.

Some protocols make explicit assumptions about the network's topology that do not match the topology we are facing here. UFlood [109] uses a base-station-rooted spanning tree on the network's topology to facilitate sender selection. A comparable tree structure is also used by Splash [110]. While laying a tree over a group of terrestrial sensor nodes distributed over a region of the surface makes sense, the linear topology of an in-line satellite formation is less suited to be represented as a tree and especially as a rooted tree,

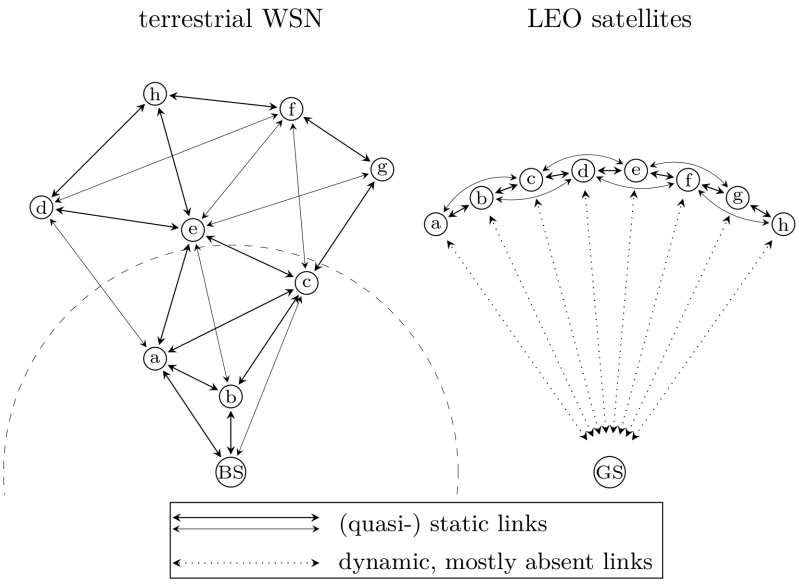


Fig. 7.2.: Schematic sketches of topologies typically assumed for terrestrial WSNs (left) and a LEO satellite formation in string-of-pearls dense orbital configuration (right).

bearing in mind that the root (the base station) is for most of the time disconnected from the rest of the network. Considering the topologies depicted in Fig. 7.2, the terrestrial network contains nodes closer to the base station (BS) like (a,b,c) that would most likely play an important role as intermediate node in OTAP and nodes farther away from the BS like (f,h). In a LEO satellite formation, all satellites have the same long-term-average distance to the GS, each of them being within a one-hop distance to the GS once in a while. Therefore, dividing satellites into intermediate and terminal nodes would be rather artificial and hardly reflect the given topology. Similarly, Sprinkler [111], CORD [112], Splash [110], and ULTRA [113] also fall into the category of protocols that we do not consider because of their assumptions about network topology.

Deluge [108], ReXOR [114], and UFlood [109] employ an end-to-end one-batch-at-a-time strategy, meaning that at any given point in time, all nodes including the base station are only transmitting messages belonging to one single batch. The base station starts to transmit messages belonging to the next batch as soon as it has received feedback indicating that dissemination of the current batch has finished in the whole network. Even though we did not verify it empirically, we deem this strategy unsuited for satellite formations, where the first hop is particularly narrow because it is constrained to short contact periods. After transmitting the first batch to some satellites within range of the GS, the GS would need to wait for the satellites to finish dissemination of the first batch to all satellites, including those currently not in range. Thus, the GS would waste the already short contact period by waiting for inter-satellite communication to finish, something that could be performed at any other time as well.

However, this chapter focuses specifically on feedback mechanisms and heuristics that intermediate nodes use to judge whether their RLNC-recoded messages are innovative for their neighbors. One may therefore ask: what are the corresponding mechanisms used by RLNC-based OTAP protocols in literature, and how do they compare to the toy protocol?

UFlood uses a heuristic approximately resembling the toy protocol's "pessimistic" heuristic introduced in Section 7.2. Even though we do not evaluate the full UFlood protocol for the aforementioned reasons, our evaluation reveals significant problems of the pessimistic heuristic in case of satellite OTAP.

COPE [115] uses opportunistic XOR coding for intermediate nodes on a per-transmission basis, as opposed to the per-batch network coding approach used by other protocols. This opportunistic use of network coding, effectively only with micro-generations of degree two, completely avoids the problem of non-innovative combinations. However, it also significantly limits the positive effects of RLNC, which we find in our evaluation to become distinct at much larger generation sizes.

AdapCode [116], Rateless Deluge [100], Synapse++[117], and SYREN [99] employ rateless codes instead of network coding: they to not recode received messages but only forward received, coded messages and/or re-encode already decoded complete batches. Arguing similarly as we did for COPE, the lack of in-network re-coding avoids any need to estimate the innovativeness of re-coded messages, but also lacks the benefits of re-coding.

A careful reader who is aware of the details of SYREN might wonder because in the paper [99] it is claimed that SYREN uses network coding and that intermediate nodes transmit only recoded messages. However, the paper also claims that SYREN's messages do not contain encoding coefficients at all, but a unique identifier from which the coefficients are pseudorandomly deduced. To the best of our knowledge, no generally applicable technique to represent the coefficients of recoded combinations in multi-hop RLNC is known. A look at the source code of SYREN reveals that indeed, re-coding does not happen and that messages of not-completed batches are only forwarded, whereas completed batches are re*en*coded.

Last but not least CodeDrip [118] does use XOR-based recoding, but is explicitly designed for rapid dissemination of small payloads and is therefore hardly comparable with the toy protocol as well as other OTAP protocols.

Finally, in one of our own co-authored publications [6] the use of rateless codes for single-hop bulk data dissemination from GSs to string-of-pearls satellite formations is discussed. In this chapter we show that while single-hop dissemination, i. e., not using inter-satellite links at all, is feasible: transmission delay as well as first-hop resource utilization can

be reduced significantly by employing multi-hop dissemination by means of network coding.

7.2 The Toy Protocol

In this section we describe the toy protocol, a minimalist tailored OTAP protocol for string-of-pearls nano-satellite formations such as the satellites of the CloudCT mission. We call it a toy protocol because we took some shortcuts to simplify protocol design. While it could in principle be implemented and used exactly as we do in our evaluation, it would be quite inflexible due to simplifications discussed at the end of Subsection 7.2.0.

7.2.0 Problem Statement

Our main design goals are:

- The protocol shall be able to achieve the goal of a complete OTAP transmission in a resource-efficient way.
- The protocol shall be as simple as possible.
- The protocol shall however be sufficiently powerful and complete so that it could in principle be deployed, violating only conventions and best practices, but not violating physical or information-theoretical bounds.
- The heuristic by which nodes decide if a satellite should further transmit messages and what RLNC chunk these messages belong to shall be easily exchangeable between different experiment runs, so that these heuristics can be compared against each other.

We define the aimed efficiency of the protocol in terms of minimizing the following metrics:

Total delay; Time to Last Byte: The time span from the GS's first transmitted message until every satellite has received and decoded the bulk payload.

Gross Inter-Satellite-Transmitted Data: The sum of the sizes of all messages transmitted by satellites, including all protocol overhead in terms of headers, encoding vectors, messages containing only control information, messages whose LC is non-innovative for every receiver, and so forth.

Consumed GS contact time: The total time that the GS spends on transmitting messages to the satellites, include time spent waiting for or receiving ACK messages.

We deliberately used the following simplifications that would need to be fixed for deployment in practice:

- As MAC protocol for inter-satellite transmissions we use a hard-coded time division multiple access (TDMA) schedule.
- All protocol parameters, including those that directly affect the size of the bulk payload to be delivered are hard-coded.
- The same holds for the satellites' formation shape: In each experiment, the number of satellites is hard-coded and the satellites "addresses" are 0 for the leading satellite through $N_{\text{satellite}} - 1$ for the trailing satellite.
- The GS contact periods are assumed to be global knowledge to the GS as well as to all satellites. Even though this data would need to be generated and communicated to the satellites in practice, we do not model the mechanisms by which this happens.
- We ignore the header conventions of the AX.25 protocol that would normally for satellite UHF links serve as the layer below the toy protocol. Instead, we use custom frame headers consisting only of 8-bit source and destination addresses (as opposed to AX.25's 112 bit) and a CRC 32 checksum.
- In one specific mode of the protocol (generous GS), we do not model the feedback mechanism by which the GS is notified that all satellites have received and decoded the whole transmission, so that it stops to transmit first hop messages. Feedback by which satellites stop to transmit, on the other hand, is part of the protocol and thus always explicitly simulated.
- We do not care about any interaction of the toy protocol with other communication demand. We assume that the task of OTAP using the toy protocol has the highest priority so that no transmission or reception of messages outside the protocol is to be modelled.

7.2.1 Protocol Overview

The protocol is on a large timescale divided into two sub-protocols. The contact period sub-protocol is applied by all nodes during formation contact periods to utilize the short and rare contact periods efficiently to upload as much data into orbit as possible. We define the term "contact period" as the intervals of time when at least one satellite of the

formation is above the GS's horizon. The other part, the inter-satellite sub-protocol, is used by the satellites for the rest of the time and handles propagation of data between the satellites.

The backbone of the data dissemination protocol is an end-wrapping overlapped chunked random linear code over a finite field $\mathbb{F}$. In our evaluation we use $\mathbb{F}_{2^8}$. The first hop messages, i. e., the transmissions of the GS, do not contain an explicit representation of the EVs. Instead, the EVs deterministically depend on the sequence number in the header, i. e., the random linear code is operated as low-overhead fountain code on the first hop. Nonetheless, in order to allow first hop messages to be later recoded, they obey the chunk-structure in the sense that each encoding vector has non-zero elements for exactly one chunk. The chunks of first hop messages are selected uniformly at random (UAR) and the EV is then drawn UAR from the corresponding chunks' subspace.

7.2.2 Contact Period Sub-Protocol

Transmissions in this regime are generally ground-station-initiated. In order to waste as little time as possible for control overhead, acknowledgments (ACKs) are used rarely.

The GS repeatedly transmits broadcast messages containing a destination address, chunk ID, sequence number, and a information vector, as depicted in Fig. 7.5. Each node receiving a message regardless of the message's destination address, processes the message by injecting the contained LC into its decoder and storing the sequence number in a ring buffer. Normally, the GS transmits messages with $\texttt{dest} = \texttt{0xFF}$, signaling that is does not request ACKs. If the GS has transmitted more than N_{unACKed} messages since the last ACK message from a satellite was received, it switches the $\texttt{dest}$ field to the address of the satellite that is angularly closest to the current GS antenna direction. If a satellite receives a GS message with a $\texttt{dest}$ matching its own address where $i := \texttt{sqn}$, it immediately responds with an ACK message that contains a bitvector encoding for each sequence number $\in [i + 1 - 2N_{\text{unACKed}}, i]_{\mathbb{Z}}$ whether the corresponding message has been received. The GS keeps requesting ACKs until an ACK message is received. When receiving an ACK message, the GS switches back to not requesting ACKs for the next N_{unACKed} messages. For a visualization of the contact period sub-protocol including this ACK mechanism, see Fig. 7.3.

Due to the ACK mechanism, the GS knows a subset of all messages that have been received by at least one satellite. The GS uses its own EAGER decoder into which it injects all messages whose successful reception was ACKed in the bitvector of an ACK message. From its own decoder, it can roughly determine whether enough messages have reached orbit to decode the whole transmission. Whether the GS should stop transmitting once

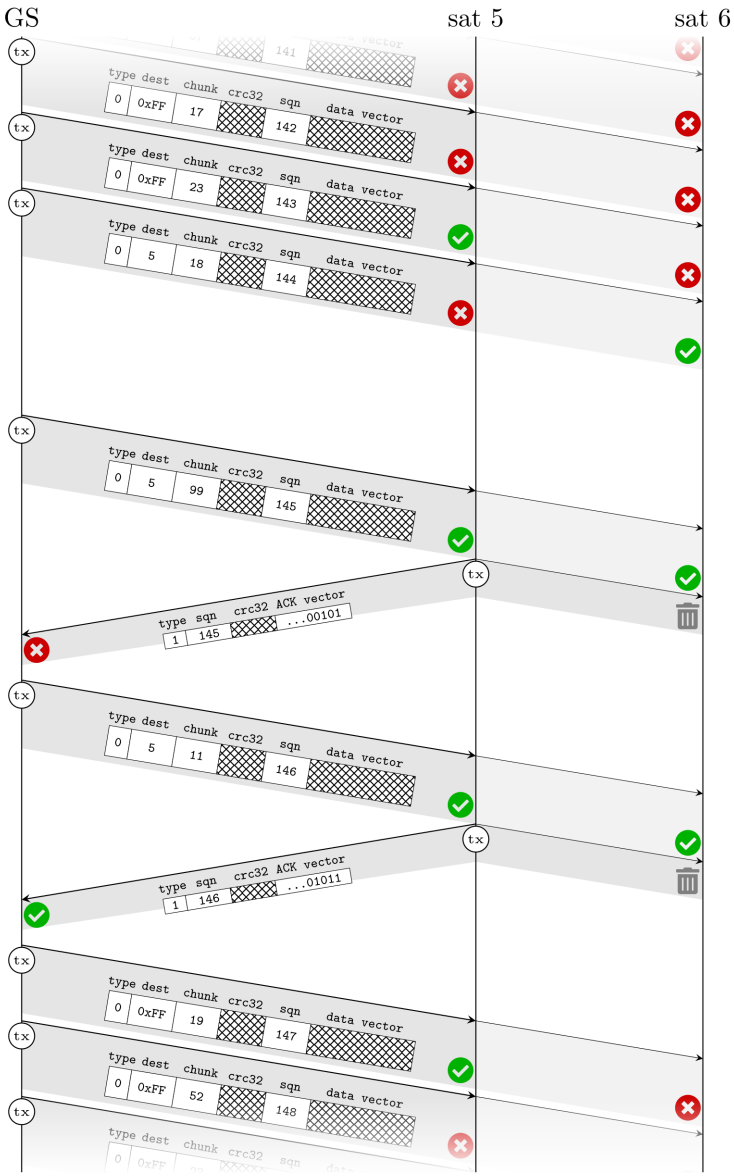


Fig. 7.3.: Schematic example of the contact period sub-protocol. Circled check marks and crosses indicate packet loss. The GS requests an ACK from satellite 5 in messages 144–146. Note that `sqn` 144 is not being ACKed, even though it was received by satellite 6. ACK requests are used rarely in order not to waste a full round-trip time (RTT) waiting after each transmission. ACK messages are ignored by other satellites.

this condition is met depends on whether the user, i. e., the satellite mission's operator, deems minimizing consumed GS contact time to be most important, or prioritizes other metrics.

Therefore, we define two operation modes for the contact period sub-protocol: in *frugal mode*, the GS stops transmitting as soon as a sufficient set of messages is ACKed to be *in orbit*; in *generous mode*, the GS unconditionally keeps pumping messages into orbit until the transmission to every single satellite is completed.

7.2.3 Inter-Satellite Sub-Protocol

Medium Access Control

This chapter focuses on RLNC protocol feedback mechanisms, so we are by no means interested in MAC here. However, in order to simulate an RLNC data dissemination protocol in a discrete event simulator for the sake of empirical evaluation, we need some kind of MAC mechanism that decides which nodes transmit at what time, hopefully suppressing primary and hidden terminal problem (HTP)-type interference. This mechanism does not need to be realistic for practical applicability. As long as we apply the very same mechanism to all protocol variants that we compare against each other, we have good reason to expect that artifacts of MAC have little influence on the qualitative aspects of our results.

Since fair channel access seems to be adequate in an equidistant string-of-pearls orbital configuration, we use a fair, fixed, hard-coded MAC schedule as a cheap trick. However, the rest of the protocol does not rely on the fixed schedule so that it could be substituted by different MAC approaches, be it TDMA or carrier sense multiple access (CSMA) style.

Time is partitioned into slots of all equal length T_{slot}: $\text{Slot}_i = [t_0 + iT_{\text{slot}}, t_0 + (i+1)T_{\text{slot}})$ with t_0 being some reference point in time (a.k.a. "Epoche"). A satellite with an address k may transmit a message in slot i if and only if

$$Q_{i \bmod |Q|} = k \bmod |Q|$$

where Q is a permutation of $[0, |Q|)_{\mathbb{Z}}$. Given that satellites' addresses are consecutive with respect to their equidistant positions in the formation, this hard-coded MAC scheme, despite being a tad inflexible, ensures

- fairness of transmission opportunities,

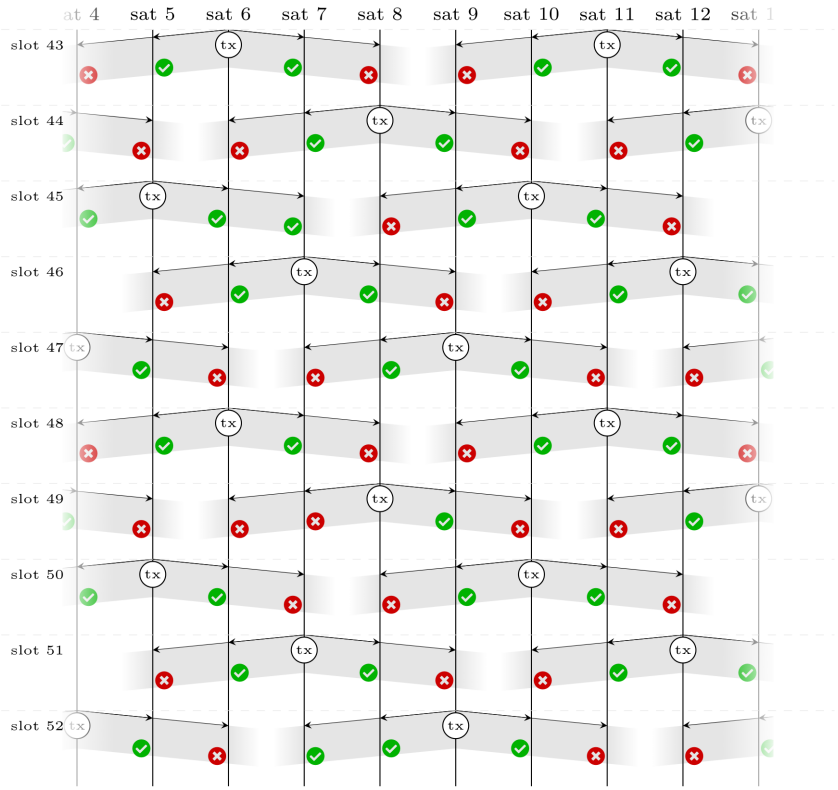


Fig. 7.4.: Example of the fixed hard-coded MAC schedule used by the toy protocol for its inter-satellite sub-protocol.

- a fixed distance of simultaneously transmitting satellites (suppressing HTP-type interference),
- and that between consecutive transmissions of one satellite, both of its direct neighbors have an opportunity to transmit a message.

In our evaluation we use $Q = (0, 2, 4, 1, 3)$, a visualization of which is given in Fig. 7.4.

Recoding Protocol

In each slot where a satellite would be allowed to transmit a coded message according to the MAC protocol, the recoding protocol decides

0. whether to include a recoded linear combination,
1. if (1), from which chunk to recode that combination,

2. what information describing its own decoder state to piggybag in the message header.

If no LC but some decoder state information is to be transmitted, a header-only message is transmitted. If neither is required, the slot remains unused by that satellite.

The message structure used by the recoding protocol is depicted in Fig. 7.5. Recoding messages are always of message type 2. The `dest` field is used only to encode "node `src` transmits this message because of its current knowledge about `dest`'s decoder state." Each of the n_{CI} chunk info fields encodes a triple (i, rank_i, h_i), corresponding to the `src`'s current decoder state.

Before describing the recoding protocol itself, we want to state and explain its desired properties:

- Nodes shall avoid transmitting linear combinations that are not innovative to any of its neighbors.
- If a node has reason to believe that for at least one chunk there is at least one neighbor, such that a randomly recoded LC of that chunk is innovative for the neighbor, a node should transmit an LC rather than stay silent.
- Transmitting an LC that is innovative for multiple neighbors is favorable over transmitting an LC that is innovative only for one neighbor.
- When the satellites reach the state that all satellites' decoders represent the same vector space, i. e., the span of all LCs received in orbit, they shall quickly and robustly fall silent. Even in the case that this decoder state is not of full rank, because there simply are not enough LCs received in orbit, we want to avoid indefinite feedback ping-pong as well as indefinitely repeated but unanswered transmissions. Every transmission costs energy as well as channel resources that could be used differently.
- We also do not want data dissemination to stall if there are at least two satellites with unequal decoder state vector space. Most importantly, we must avoid an unfinished dissemination stall if the set of LCs received in orbit is sufficient to decode the whole transmission. The last point is of paramount importance, because it leads to an unrecoverable deadlock if the GS uses frugal mode.

Chunk selection Each node maintains a neighbor table containing the rank and possibly decoder state hash of each chunk for each neighbor node. A node a computes for each neighbor b and each chunk i the *decoder superiority* $\eta^i_{a,b}$ as a rough estimate for the lower bound of the number of innovative LCs of chunk i that node b can receive from node a and

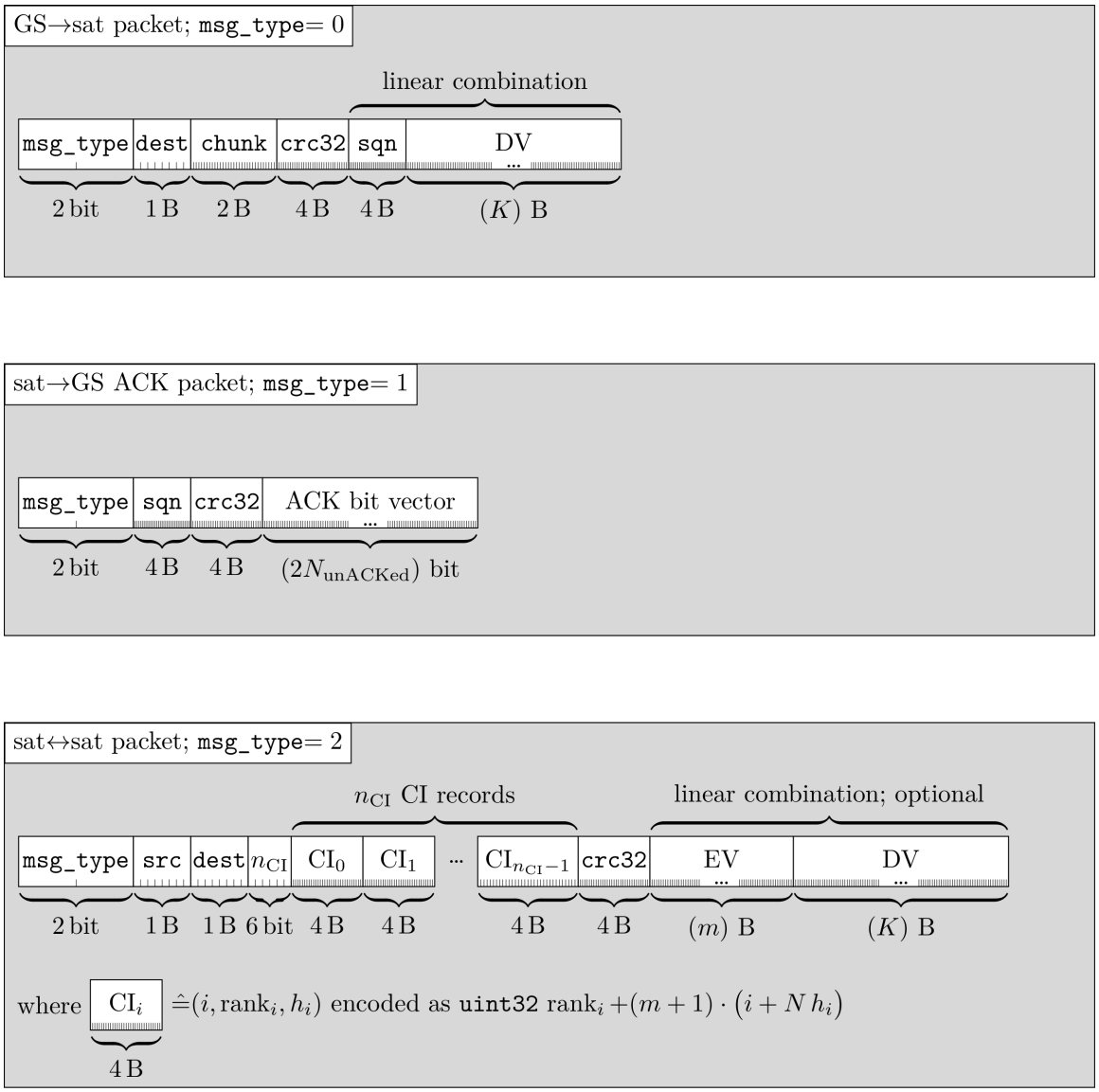


Fig. 7.5.: Message structure of the toy protocol's three different message types. The parts' sizes are given in units of bit and byte (B).

inject. From the decoder superiorities, a per-chunk fitness value is computed according to (7.0).

$$\varphi_a^i = \sum_{b \in \text{neighbors}(a)} f(\eta_{a,b}^i) + \text{feedback urgency term} \tag{7.0}$$

$$\text{where } f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 2 - \frac{1}{x} & \text{else} \end{cases}$$

By using the non-linearly saturating but monotonically increasing function f to rescale $\eta_{a,b}^i$, we achieve that a chunk with two neighbors of decoder superiority ≥ 1 always has higher fitness than a chunk with only one such neighbor.

Decoder state feedback Each node maintains two sets of chunk indices called *regular feedback set* and *urgent feedback set*. Whenever the node's decoder rank corresponding to a chunk changes (i. e., increases), the corresponding chunk index is added to the regular feedback set. Whenever a node a receives a non-innovative LC of chunk i from a node b in a message with $\texttt{dest} = a$ and $\varphi_a^i < 1$, this necessarily means that node b's neighbor table with regard to node a and chunk i is outdated. Therefore, i is added to the urgent feedback set. Elements of the urgent feedback set are always immediately cleared from the regular feedback set.

In each slot where a node is allowed to transmit according to the MAC schedule, a sequence of up to $n_{\text{CI}}^{\max}$ records, i. e., triples (i, rank_i, h_i), is included in the message header. If a chunk i is selected (according to the chunk fitness rules) for transmission of an LC, i is used for the first chunk info (and cleared from both feedback sets). Then, chunk indices for the remaining chunk info records are popped from the urgent feedback set and then from the regular feedback set, until either these sets are emptied or the desired maximum per-message number $n_{\text{CI}}^{\max}$ has been selected.

If not a single chunk info record is to be transmitted (which is only the case if no LC is to be transmitted as well), no message will be transmitted. If there is no chunk selected for transmission of an LC but chunk info records need to be transmitted, the message will consist of the header only.

This feedback mechanism based on chunk info records ensures that every change of a satellite's decoder state will be announced in one transmission. If that message gets lost, the rules of urgent feedback ensure that this information will be retransmitted in the following messages until transmissions of non-innovative LCs originating from message loss are being suppressed.

We use

$$(\text{feedback urgency term})_i = \begin{cases} \frac{1}{2N^2} & \text{if } i \text{ in any feedback set} \\ 0 & \text{else} \end{cases}$$

in (7.0). This means that if chunk selection according to only decoder superiority values yields more than one fittest chunk, we break ties so that chunks requiring feedback is required for are favored.

Remaining header fields The `src` and `num_chunk_infos` header fields are trivially set to the node's own address and the number of chunk info records encoded in the message. If the message from $\texttt{src} = a$ includes an LC from chunk i, `dest` is set to the address of the neighbor b that maximizes $\eta^i_{a,b}$, breaking ties randomly. Thus inconsistencies in the satellites' neighbor tables are robustly revealed.

Decoder superiority heuristics Last but not least we have not yet discussed, what heuristic is used by nodes to compute $\eta^i_{a,b}$ from the own decoder state and the neighbor table.

Using the chunk ranks as only input variables, we use bounds that directly follow from linear algebra in case of ranks being unequal: a random element of a subspace of dimension $\text{rank}_a(i)$ is with high probability not inside a subspace with smaller dimension $\text{rank}_b(i)$. Since injecting a single LC can increase a decoder's rank at most by one, it follows that at least a sequence of $\text{rank}_a(i) - \text{rank}_b(i)$ random LCs from a can be innovatively injected into b.

$$\eta^i_{a,b} = \begin{cases} \text{rank}_a(i) - \text{rank}_b(i) & \text{if } \text{rank}_a(i) > \text{rank}_b(i) \\ 0 & \text{else} \end{cases} \tag{7.1}$$

We call (7.1) the *pessimistic decoder superiority heuristic*. For this heuristic we can anticipate one significant problem: considering two neighboring nodes, this heuristic allows only the node with greater chunk rank to transmit LCs. Since every innovative LC increases the receiver's rank by 1, this process directly leads to the situation where both decoders have the same rank, then no further transmission can happen. Since different linear subspaces can have equal rank, this can lead to the situation that we definitely want to avoid: for an arbitrary chunk, all nodes in the network reach the same (but not full) rank. Due to the pessimistic heuristic, no messages belonging to this chunk are transmitted anymore, and the dissemination stalls, regardless of whether a full rank could be achieved or not. We call this phenomenon the "pessimistic stall."

An obvious solution to avoid this is to offset the pessimistic heuristic by one, resulting in the *optimistic decoder superiority heuristic* (7.2).

$$\eta_{a,b}^{i} = \begin{cases} 1 + \text{rank}_a(i) - \text{rank}_b(i) & \text{if } \text{rank}_a(i) \geq \text{rank}_b(i) \text{ and } \text{rank}_b(i) < m \\ 0 & \text{else} \end{cases} \tag{7.2}$$

While this successfully avoids stalling of data dissemination, it leads to the complementary problem that in a situation where all nodes' decoders represent equal linear subspaces not being of full rank, all nodes still keep on transmitting messages that are useless, because the contained LCs are non-innovative to every node.

Therefore, we propose the *hash-based decoder superiority heuristic* (7.3), that equals the pessimistic heuristic for unequal ranks and judges based on decoder state hash values in case of equal rank.

$$\eta_{a,b}^{i} = \begin{cases} \text{rank}_a(i) - \text{rank}_b(i) & \text{if } \text{rank}_a(i) > \text{rank}_b(i) \\ 1 & \text{if } \text{rank}_a(i) = \text{rank}_b(i) \text{ and } h_a(i) \neq h_b(i) \\ 0 & \text{else} \end{cases} \tag{7.3}$$

While this requires some overhead, namely a decoder capable of computing a decoder state that uniquely depends on the represented subspace, as well as including hash values of that decoder state in the message headers, it solves both of the aforementioned problems.

Of course one could conceive a plethora of different methods to fix these problems without resorting to hash values, but instead to propagate the information that a non-innovative LC has been received back to the transmitter:

- Using the optimistic heuristic, nodes receiving a non-innovative LC from an equal-rank neighbor could notify the neighbor of this fact.

- Using the pessimistic heuristic, nodes stalling transmissions due to all neighbors' chunk ranks being equal to their own could transmit probe messages, containing only encoding vectors, to overcome indefinite stalling.

In parallel to the development of EAGER we have played around with such protocol variants but did not get evaluation results that were promising enough in comparison to the hash-based heuristic. Therefore, and because every of these variants adds another layer of protocol complexity, we refrain from describing them in detail or including them in the evaluation presented here.

7.3 Evaluation

To evaluate the usefulness of a unique and thereby hashable decoder state in RLNC used for OTAP, we compare different variants of the protocol described in the last section against each other. We consider scenarios with a number of satellites in string-of-pearls orbital configuration with 66° inclination, 600 km orbital altitude, and 100 km orbital distance between each pair of adjacent satellites. As goal of each simulation run we consider the broadcast of one bulk file, e. g., a firmware image, from a single GS to each satellite in the formation.

The complete and error-free reception of the transmission payload at every satellite can be considered the non-negotiable main objective. Depending on the satellite mission, an operator might seek to optimize the transmission process with respect to different metrics. In each simulation run, we measured each of the following secondary metrics:

Time to last byte The total time span from the beginning of transmission of the first message from the GS until each satellite has decoded the full bulk transmission. A satellite operator will update the firmware for a certain reason, e. g., a bug-fix, and will likely want the new firmware version to be used as soon as possible. If the GS is instructed to transmit the bulk file at a point in time when no satellite is in contact range, it will need to wait until the beginning of the next contact period. Since neither GS-to-satellite nor inter-satellite transmissions related to the OTAP goal can take place during this initial waiting time, it is excluded from the time to last byte measured.

Consumed GS contact time In transmitting OTAP messages to satellites, the GS consumes channel resources that otherwise could be used for different tasks, like telemetry or command. For each message transmitted by the GS, we count the transmission time and, if the transmitted message contains an ACK request, the time that the GS waits for a potential ACK response message from the satellite.

Number of messages and number of bytes transmitted in orbit For each message transmitted by a satellite that is not an ACK to the GS, we count the message as well the total number of bytes the message consists of, including all headers and check sums. The total number of bytes transmitted can serve as a proxy for the energy consumed by satellites for the OTAP objective as well as for the amount of channel resources consumed that could otherwise have been used for different inter-satellite communication purposes. However, optimizing an OTAP protocol for this secondary metric alone does not make sense in the context of evaluating RLNC-based multi-hop OTAP techniques: in the scenarios that we consider, every satellite is in principle a direct neighbor of the GS and therefore a single-hop transmission,

e. g., using a rateless code, is sufficient to achieve OTAP without inter-satellite communication at all.

7.3.0 Orbital and Geographical Scenario Setup

As foundation for our evaluation, we modeled the scenarios in the ESTNeT [56] simulator. Each simulation's topology consists of one satellite ground station located at Würzbug (Germany) and equipped with a transmitter operating at $15\,\mathrm{W}$ transmit power and a steerable high-gain Yagi antenna with $21°$ half-power beam width (HPBW) as well as ten satellites in a string-of-pearls configuration, each equipped with a transceiver operated at $2.5\,\mathrm{W}$ transmit power and an idealized omnidirectional antenna. Packet loss was simulated using the signal-to-interference-plus-noise-based probabilistic error model of ESTNeT's APSK radio channel. For background noise power we implemented and used an isotropic scalar model in which noise depends on the receiving satellite's position according to data gathered by the UWE-3 mission in 2014 for a center frequency of $437.2\,\mathrm{MHz}$ [107].

All metrics that we measure strongly depend on the lengths of GS-to-satellite contact periods, the corresponding maximum elevation angles, et cetera. In order not to choose the absolute time of our experiments and the orbits' right ascension of the ascending node (RAAN) arbitrarily, we adjusted the orbital elements so that the geometrical center point of the formation passes zenithally over the satellite GS during the first contact period. As geographical location of the satellite GS we use the University of Würzburg, where the UHF satellite GS of the UWE-1 through UWE-4 satellite missions is operated.

During each contact period, the GS does not track (i. e., point its high gain antenna at) individual satellites but instead sweeps its antenna over the formation in a fashion that we call sweep-tracking.

Where not stated differently, each data point in each plot within this chapter corresponds to ten independently seeded simulation runs and error bars depict $95\,\%$ confidence intervals.

7.3.1 Sweep Tracking

Given a string-of-pearls configuration of N_{sat} equidistantly spaced satellites numbered consecutively from 0 for the leading satellite through $N_{\mathrm{sat}} - 1$ for the trailing satellite, let $[t_{\mathrm{b}}, t_{\mathrm{e}}]$ be a formation contact period where satellite i_{b} rises at t_{b} and satellite i_{e} sets at t_{e}. For typical contact periods we will just have $i_{\mathrm{b}} = 0$ and $i_{\mathrm{e}} = N_{\mathrm{sat}} - 1$, but different values

with $i_\mathrm{b} \leq i_\mathrm{e}$ are possible if either the contact period's maximum elevation is very small or the formation is very long.

If $i_\mathrm{b} = i_\mathrm{e}$, we simply fall back to tracking that individual satellite alone. Otherwise, we compute

$$\tau(t) := i_\mathrm{b} + (i_\mathrm{e} - i_\mathrm{b}) \cdot s\left(\frac{t - t_\mathrm{b}}{t_\mathrm{e} - t_\mathrm{b}}\right) \quad \text{for } t \in [t_\mathrm{b}, t_\mathrm{e}] \tag{7.4}$$

where s is a continuous scaling function $s : [0,1] \to [0,1]$ satisfying $s(0) = 0$ and $s(1) = 1$. For sake of simplicity we can for now assume s to be the identity function. In this trivial case, τ just linearly maps the interval $[t_\mathrm{b}, t_\mathrm{e}]$ to the interval $[i_\mathrm{b}, i_\mathrm{e}]$. In the GS's horizontal coordinate system, let $d_i(t)$ be the function that yields the direction in which the satellite i is located at time t and let $d_\mathrm{antenna}(t)$ be the direction of the GS's antenna at time t. Then, sweep tracking is defined by (7.5).

$$d_\mathrm{antenna}(t) = \begin{cases} d_{\tau(t)}(t) & \text{if } \tau(t) \in \mathbb{Z} \\ \mathrm{Slerp}\Big(d_{\lfloor \tau(t) \rfloor}(t), d_{\lfloor \tau(t) \rfloor + 1}(t); \mathrm{frac}(\tau(t))\Big) & \text{else} \end{cases} \tag{7.5}$$

This means, when $\tau(t)$ happens to be integer, satellite $\tau(t)$ is pointed to directly, and between these points in time, the antenna direction is gained by interpolating between the adjacent satellites' directions by means of spherical linear interpolation.

In our evaluation we use a simple one-parameter family of functions

$$s_\sigma(\tau) = \tau + \sigma \frac{\sin(2\pi\tau)}{2\pi} \quad \text{with } \sigma \in [-1, 1]$$

where we complement the linear identity function part with an additive sinusoidal component resulting in more horizon-focused ($\sigma < 0$) or zenith-focused ($\sigma > 0$) sweep tracking.

The goal and effect of sweep tracking is similar to the method that we published in [6], which was, because of lack of communication, developed and implemented independently at the same time and for the same purpose.

7.3.2 External Boundary Conditions

There are a number of boundary conditions that we have to assume to set up our experiments. These are *external* in the sense that they are given by the satellite mission itself and can therefore hardly be influenced in order to improve OTAP performance. In order to broaden the scope of our results, we varied some of them across different experiments:

Number of satellites We varied the number of satellites in the formation between ten, which corresponds to the number of satellites in the CloutCT mission [106], and 50, which outnumbers any string-of-pearls configuration satellite mission we are aware of. Since a formation of ten satellites is currently of greater practical relevance considering currently operated and planned multi-satellite missions, and the results corresponding to greater numbers of satellites do not bring any additional insights qualitatively, all figures presented in the work correspond to ten satellites.

Bulk transmission payload size The size of the payload to be broadcast to every satellite was varied between 300 kB and 3 MB, leading to transmissions ranging from requiring only a fraction of one contact period and no inter-satellite communication to requiring multiple GS overflights.

Other external conditions where not varied; instead, reasonable values where chosen. This includes the orbital parameters, the geographical location of the GS, the GS's and satellites' antennas, the noise and error model, and the channel carrier frequency, modulation, and baud rate.

7.3.3 Protocol Parameters

From the protocol design perspective there is also a great number of parameters that can and have to be chosen in order to transmit a bulk file using RLNC, namely the

- chunk selection heuristic (optimistic/pessimistic/hash-based)
- transmission strategy of the GS (frugal or generous)
- finite field (order as well as generating polynomial) over which the code is implemented
- chunk size m
- chunk overlap ℓ
- number of chunks N
- information vector size K
- value range of the hash function used to compare decoder states $[0, h_{\text{bound}})_{\mathbb{Z}}$
- maximum number of chunk info records per message $n_{\text{CI}}^{\text{max}}$
- sweep tracking parameter σ

The one parameter that we are most interested in is the chunk selection heuristic, because if we can answer which chunk selection heuristic leads to the best OTAP performance, we know whether being able to compare decoder states based on hash values yields any benefit.

The GS's transmission strategy is adjusted differently according to which metrics we look at. When looking at the time to last byte, it makes sense to let the GS transmit as many combinations as possible until the entire transmission process has finished. If the goal however is to minimize the GS contact time used for the transmission, we put the GS into frugal mode, meaning that it falls silent as soon as it knows that the set of all messages received in orbit is sufficient to decode the coded bulk transmission.

Even though we were in principle able to simulate the transmission of messages of arbitrary size, the probabilistic error model implemented by ESTNeT makes message loss probability explicitly depend on the message size. To make things worse, this error model does not include the burstiness of noise, implying that the packet-size-dependence of message loss probabilities is hardly realistic in an environment where a significant share of background noise can be assumed to originate from other systems' radio transmissions (i. e., interference of nodes not modelled) and military radar systems. To limit the influence of these effects on our simulation result, we try to keep the message sizes within a small window across all experiments. We use a constant but feasible maximum message size of $S = 256\,\mathrm{B}$ including all headers, and set the other parameters accordingly.

As finite field we use an order of 2^8 so that each byte of payload can be encoded exactly with one finite field element.

All other parameters' effects on OTAP performance were investigated empirically. However, the other parameters cannot be chosen independently, but must meet certain requirements in order to respect the needs of the transmission: chunk size and information vector size must together be small enough that the maximum total message size is not exceeded; the number of chunks must be large enough to allow the whole payload to be encoded.

Apart from the sweep tracking parameter σ, we use the following set of independent parameters to determine chunk size, chunk overlap, number of chunks, information vector size, and value range of hash function.

serialized chunk info record size To encode a chunk info record in the header, instead of using fixed size fields for chunk id, chunk rank, and chunk hash, we use one unsigned integer of size w_{CI} that encodes these three numbers. Since chunk rank values can only lie in $[0, m]_{\mathbb{Z}}$, chunk id values lie in $[0, N)_{\mathbb{Z}}$, we encode a chunkinfo record as

$$\mathrm{CI}_i^{\mathrm{serialized}} := \mathrm{rank}(i) + (m+1) \cdot (i + N h_i) \tag{7.6}$$

which is smaller than $2^{w_{\text{CI}}}$ and can therefore be encoded as w_{CI} bit unsigned integer as long as

$$h_i < \left\lfloor \frac{2^{w_{\text{CI}}}}{(m+1) \cdot N} \right\rfloor = h_{\text{bound}}. \tag{7.7}$$

Maximum number of chunk info records per message Since each message consists of (see Fig. 7.5) 3 bytes for source address, destination address, and number of chunk infos plus no more than $n_{\text{CI}}^{\max} \cdot w_{\text{CI}}/8$ byte for the chunk infos, there are $S' := S - 3\,\text{B} - n_{\text{CI}}^{\max} \cdot w_{\text{CI}}$ bit left for linear combinations, i. e., for encoding vector and information vector.

Relative encoding vector size The encoding vector of a linear combination equals the full chunk size m and therefore requires m bytes in the message if a finite field of order 2^8 is used. For given $e^{\text{r}} \in [0, 1)$ we compute $m = \max\{1, \lfloor e^{\text{r}} \cdot S'/\text{B} \rfloor\}$. The remainder of the message can be used for the combination's information vector: $K = S'/\text{B} - m$.

Relative chunk overlap As with the full chunk size, we specify the overlap of consecutive chunks as a relative quantity $\ell^{\text{r}} \in [0, 1)$ that expresses the ratio of overlap and full chunk size: $\ell = \lfloor \ell^{\text{r}} \cdot m \rfloor$

Bulk payload size The ultimate goal of the whole system is to transport a bulk payload of total size $\mathcal{S}$. The system of linear equations comprising all chunks has $(m - \ell) \cdot N$ rows and each row of the solution carries K byte of uncoded payload data. Therefore, we choose the minimum number of chunks required to encode all payload data: $N = \left\lceil \frac{\mathcal{S}}{(m-\ell) \cdot K\text{B}} \right\rceil$

For the serialized chunk info record size we just use $w_{\text{CI}} = 32\,\text{bit}$. After some experiments, we settled for $e^{\text{r}} = 0.2$, $\ell^{\text{r}} = 0.1$, $n_{\text{CI}}^{\max} = 6$, $\sigma = 0.25$. To justify this choice, we start the discussion of the experiments' results by examining the influence of changing each of these four parameters, one by one.

We want these parameters to be equal when comparing the different heuristics against each other. Therefore, we first try to set up a scenario where inter-satellite recoding is crucial to complete the transmission, but where we hope that neither pessimistic stall nor optimistic non-innovative babble occurs.

In order for inter-satellite recoded transmissions to be needed at all, it is crucial to choose a payload large enough that not all satellites can decode the transmission already from the messages that the GS transmitted during the first formation contact period.

Therefore, we measured for each satellite the number payload bytes received during the first contact period when the GS is transmitting generously, i. e., not frugally.

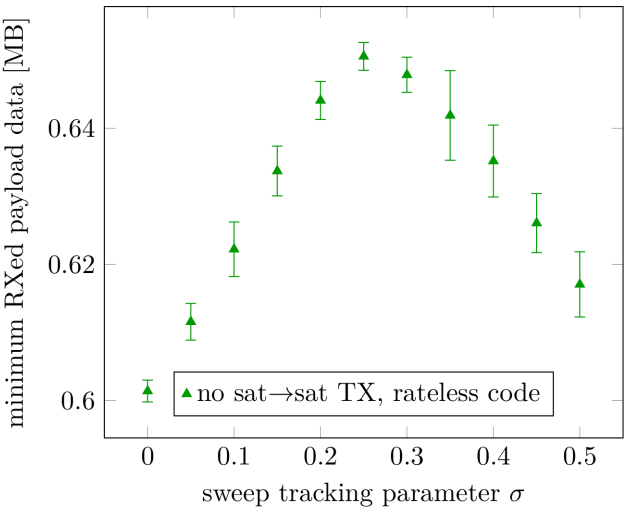


Fig. 7.6.: Minimum per-satellite single-hop goodput during first contact period depending on sweep tracking parameter σ.

In Fig. 7.6 we plotted the minimum of this number across all satellites over the sweep tracking parameter σ. This roughly corresponds to the maximum amount of data that can be transmitted by the GS to all satellites within the first contact period using rateless coding without any inter-satellite communication. The data reveals that the maximum payload size that can be transmitted during one overflight, leading to a completed transmission before the satellites even have a chance to exchange recoded messages, lies somewhere around 650 kB. For our tuning experiments we chose a slightly larger payload of 750 kB to ensure that some inter-satellite communication will be needed to finish the transmission before the start of the next contact period. In addition, we could verify that this is still small enough that the entire transmission is *in orbit* after the first contact period, therefore avoiding the problems of the optimistic heuristic from growing indefinitely.

As we show later, the pessimistic stall problem occurs with a much higher probability when the GS is operated in frugal mode. Therefore, we conducted the parameter-tuning experiments in generous GS mode.

Let us first have a look at the sweep tracking parameter σ. Using a generous GS and a bulk payload size of 750 kB, we varied $\sigma \in [-1, 1]$ and plotted the time to last byte as well as the total number of bytes transmitted in inter-satellite communication over σ in Fig. 7.7. The value of $\sigma = 0.25$ seems to be a reasonable choice because both metrics seem to have an optimum somewhat near that value and neither metric worsens steeply on either side.

Next we investigate the influence of e^{r}, the relative share of encoding vectors in the linear combination part of a message. Fig. 7.8 shows the time to last byte as well as the total

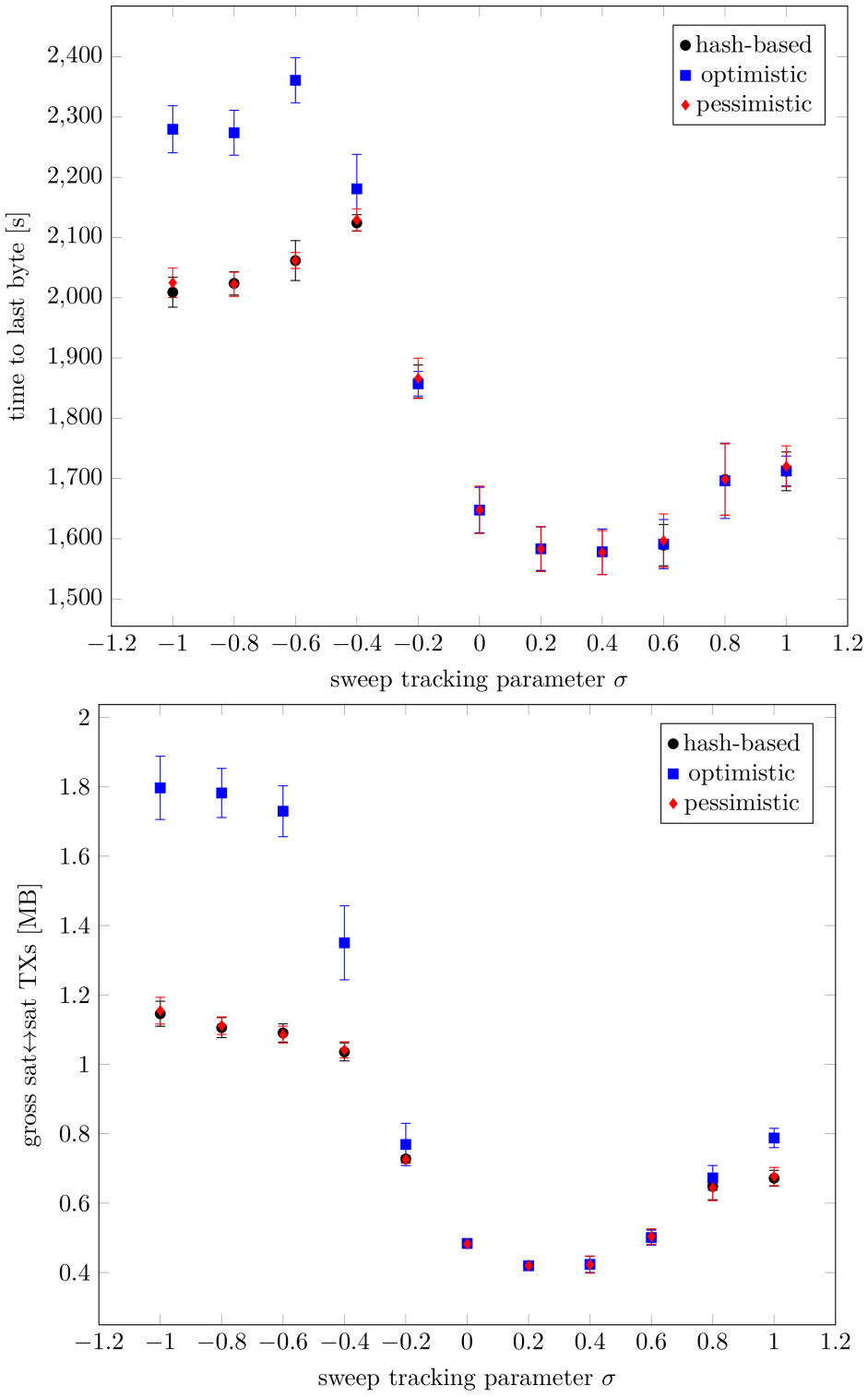


Fig. 7.7.: Time to last byte (top) and total inter-satellite transmissions (bottom) depending on sweep tracking parameter σ for each heuristic, 750 kB bulk payload and generous GS transmission strategy.

number of bytes transmitted in inter-satellite communication, each plotted over e^{r} for a generous GS and a bulk payload size of 750 kB. The gross inter-satellite transmissions show a rather shallow optimum somewhere in the range $e^{\mathrm{r}} \in [0.20, 0.35]$. Without changing the total message size, this parameter literally encodes the relative share of encoding vectors, i. e., RLNC-induced overhead in each message, It is quite remarkable that the total resource consumption hardly depends on the on this overhead for $e^{\mathrm{r}} \in [0.2, 0.35]$, meaning that even nearly doubling the overhead to 35 % seems to be entirely compensated by the benefits of re-coding. Since the transmission delay seems to worsen significantly for $e^{\mathrm{r}} > 0.2$, we chose a value of $e^{\mathrm{r}} = 0.2$ for the rest of this evaluation.

ℓ^{r} is the overlap of successive chunks in units of encoding vector size. The data shown in Fig. 7.9 suggest that the protocol performance depends only weakly on this parameter. Both the total inter-satellite transmissions and the time to last byte seem to have an optimum located somewhere at $\ell^{\mathrm{r}} \in [0.05, 0.2]$. However, the metrics' variation within that range is so small compared to the corresponding standard deviation that it is hard to make out a robust optimal value. Therefore, we just note that the technique of overlapped chunked RLNC does not after all seem to offer overwhelming benefits in our toy protocol and settle for a value of $\ell^{\mathrm{r}} = 0.1$ for the remainder of this work.

The last thing we varied was $n_{\mathrm{CI}}^{\max}$, the maximum number of chunk info records included per message. Requiring a maximum for this number at all can be seen as an artifact of using slotted MAC. The slots' length enforces a maximum message size (that we chose to be 256 B as stated earlier) and thereby the maximum number of chunk info records per message directly influences the size of the encoding vector as well as information vector. Before we measure the impact of this quantity, we want to reason about a useful choice. For a satellite in the middle of the formation, we would expect that, under good conditions, the satellite receives innovative LCs from its direct neighbors on both sides between two successive transmissions. Each innovative reception results in a chunk rank change that needs to be published. Assuming that it is not unlikely for the transmitter to select a different chunk for transmission, there are already three chunks from which chunk info records are to be transmitted. A requirement for additional chunk info record can occur for three reasons: retransmission of previously transmitted chunk info records can be implicitly requested using the `dest` address; messages received from non-nearest neighbor nodes can change chunk ranks as well; in case of non-vanishing overlap, EAGER's decoding can ripple through, in principle, an arbitrary amount chunks. In fact, it can happen that before the last innovative message is received, every chunk has a rank of $m - 1$ and each non-empty row of the decoder has two non-zero elements, so that the next innovative message, regardless of which chunk it belongs to, finalizes the transmission and effectively increases every chunk's rank by 1.

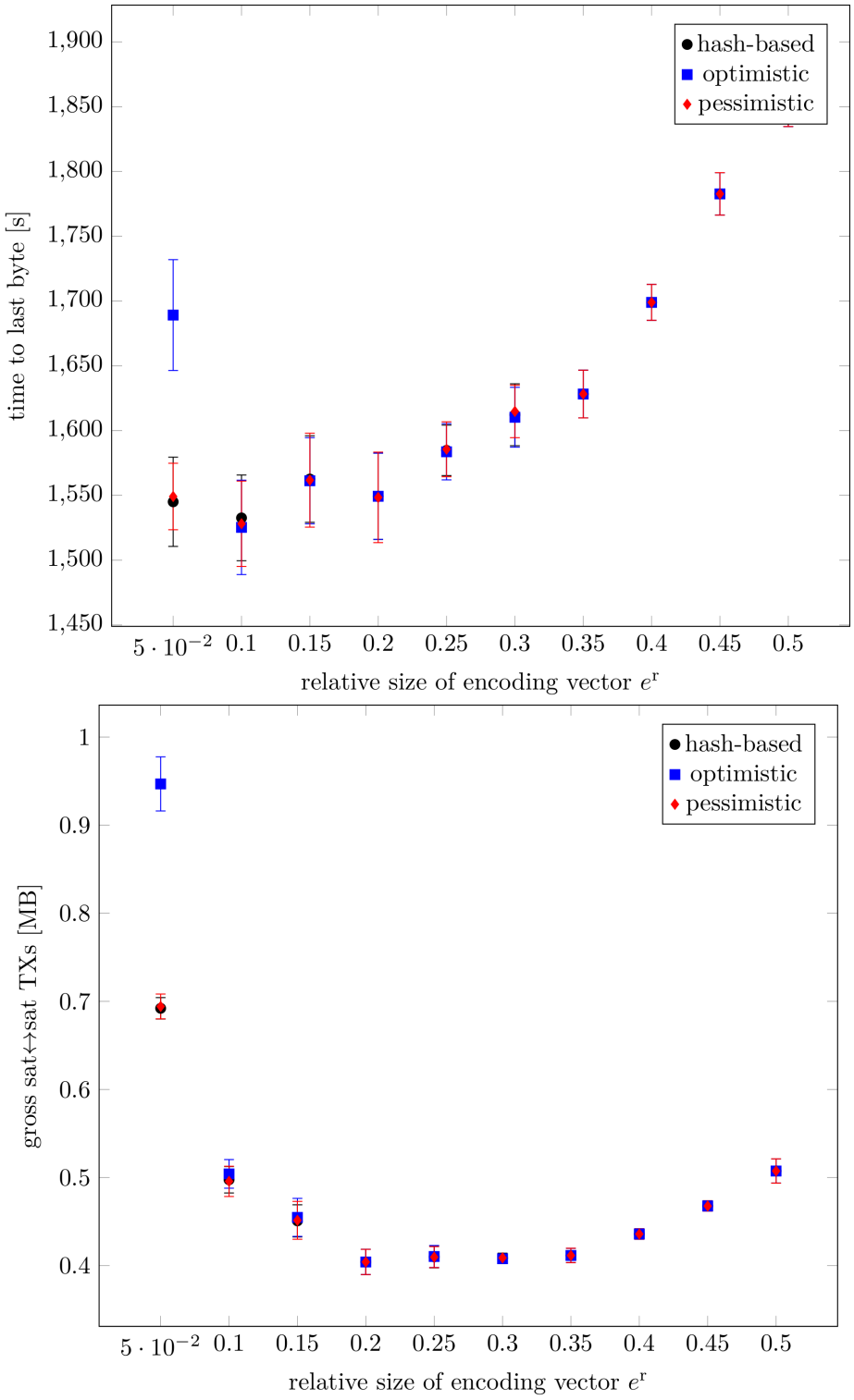


Fig. 7.8.: Time to last byte (top) and total inter-satellite transmissions (bottom) depending on relative size of encoding vectors e^{r} for each heuristic, 750 kB bulk payload and generous GS transmission strategy.

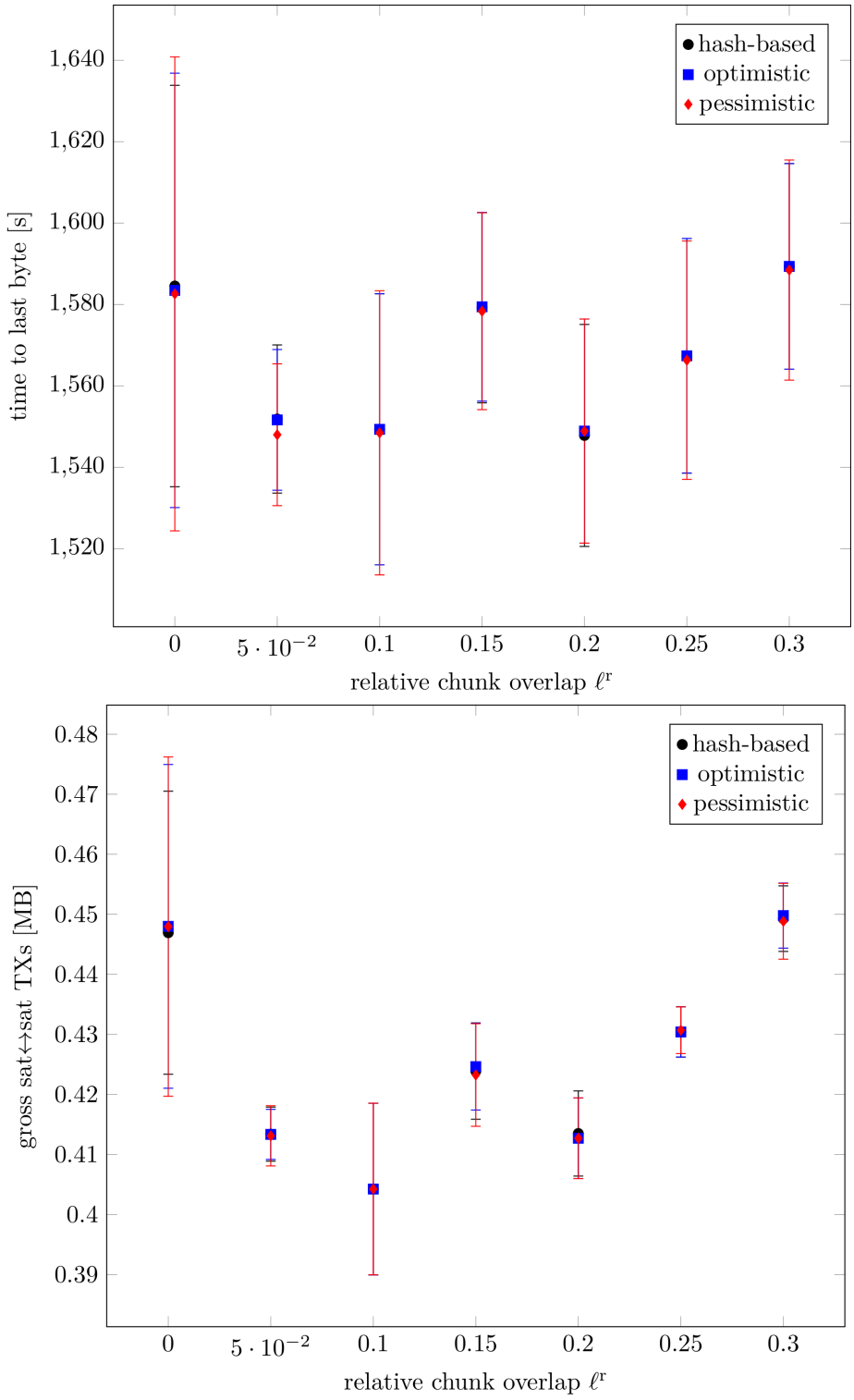


Fig. 7.9.: Time to last byte (top) and gross inter-satellite transmissions (bottom) depending on relative chunk overlap ℓ^{r} for each heuristic, 750 kB bulk payload and generous GS transmission strategy.

So we expect that performance would drop for $n_{\text{CI}}^{\max} < 3$. On the other end of the spectrum, we would expect the number of inter-satellite transmitted bytes to rise measurably as soon as the message header takes up a macroscopic share of the message size. The header of a typical 3-chunk-info message has 20 B, so such a header makes up less than $10\,\%$ for $n_{\text{CI}}^{\max} = 3$ and $20\,\%$ for $n_{\text{CI}}^{\max} = 37$. From that we would expect a very weak decline in protocol performance as $n_{\text{CI}}^{\max}$ grows to larger two-digit numbers. The results shown in Fig. 7.10 confirm exactly this expectation, so we somewhat arbitrarily chose $n_{\text{CI}}^{\max} = 6$ for the remainder of this evaluation, as this value is deep in the near-optimum region and far away from values degrading protocol performance.

7.3.4 Comparison of Heuristics

The main question that we want to answer in this chapter is: "Does the ability to compare RLNC decoders by means of a hash value offer any performance benefits for OTAP in string-of-pearls-configuration satellite formations?" The answer is "Yes!", as we see in this subsection.

We conducted all experiments for parameter tuning in a setting of 10 satellites and a bulk payload size of 750 kB that we chose specifically to avoid the anticipated weaknesses of the pessimistic and optimistic heuristic. When a satellite operator wants to use the protocol not only for benchmarking, but to actually deliver a useful payload to the satellites, the size of that payload is an externally given constraint.

Therefore, we now stop to scan through protocol parameters and instead vary the payload size in a range of 300 kB to 3 MB. We have already seen during parameter tuning that the optimistic heuristic performs significantly worst. When going to larger payload sizes that require multiple contact periods for upload, the optimistic heuristic results in all satellites permanently forwarding non-innovative LCs between GS contact periods, not only wasting resources but also leading to quite expensive simulations. Therefore, we did not include the optimistic heuristic in the following experiments.

In Fig. 7.11 we depict the performance of the protocol variants using different heuristics for a generous GS and varying bulk payload size. Interestingly, the pessimistic heuristic seems to marginally outperform the hash-based heuristic, even though the benefit is hardly significant from a practical point of view.

This picture slightly changes if we run the GS in frugal mode, meaning that it stops to transmit as soon as a sufficient set of messages has been ACKed by the satellites. As can be seen in Fig. 7.12, the pessimistic and hash-based heuristic are still roughly on par quantitatively. However, with some finitely small probability, the anticipated

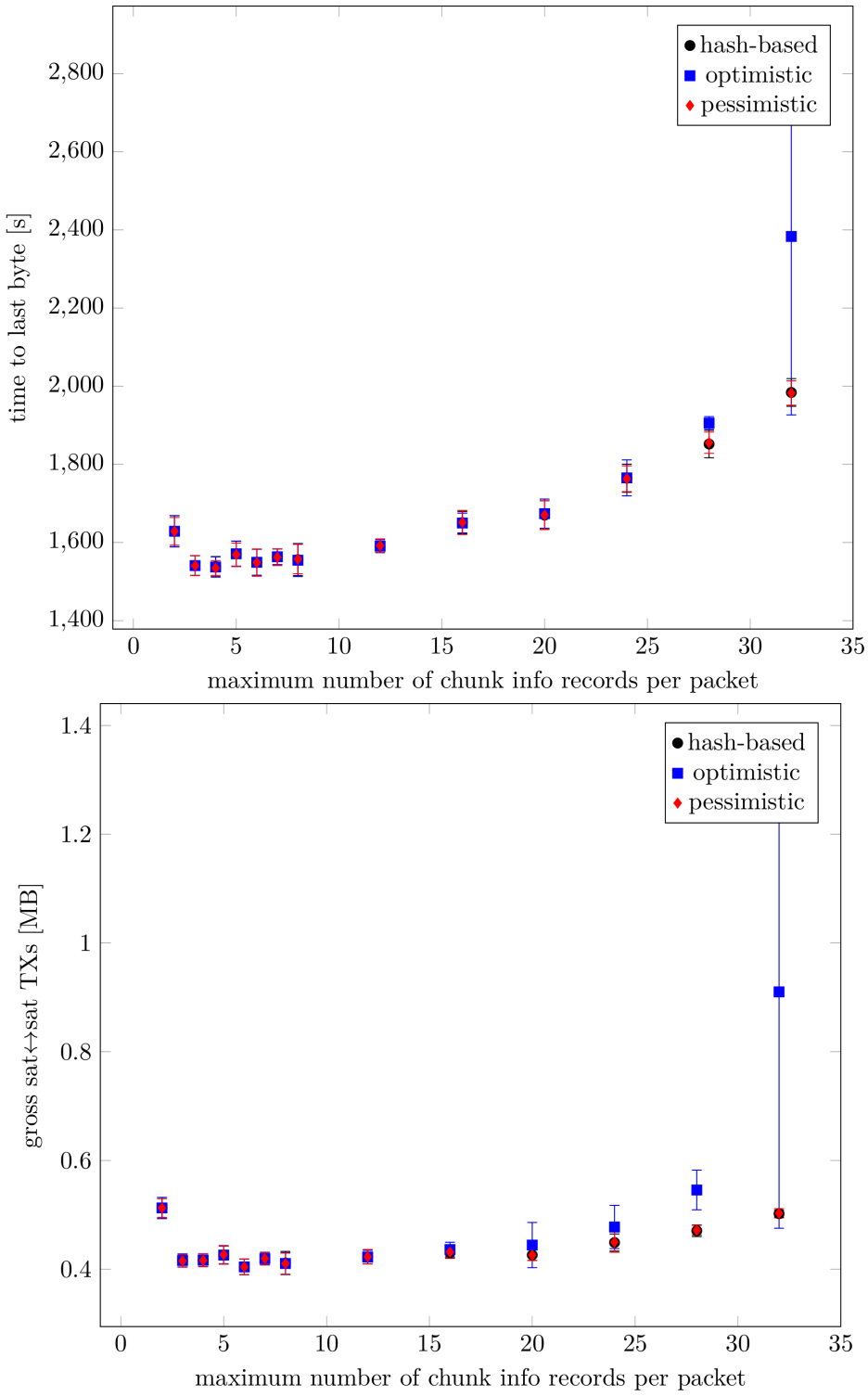


Fig. 7.10.: Time to last byte (top) and total inter-satellite transmissions (bottom) depending on maximum number of chunk info records per message $n_{\mathrm{CI}}^{\max}$ for each heuristic, 750 kB bulk payload and generous GS transmission strategy.

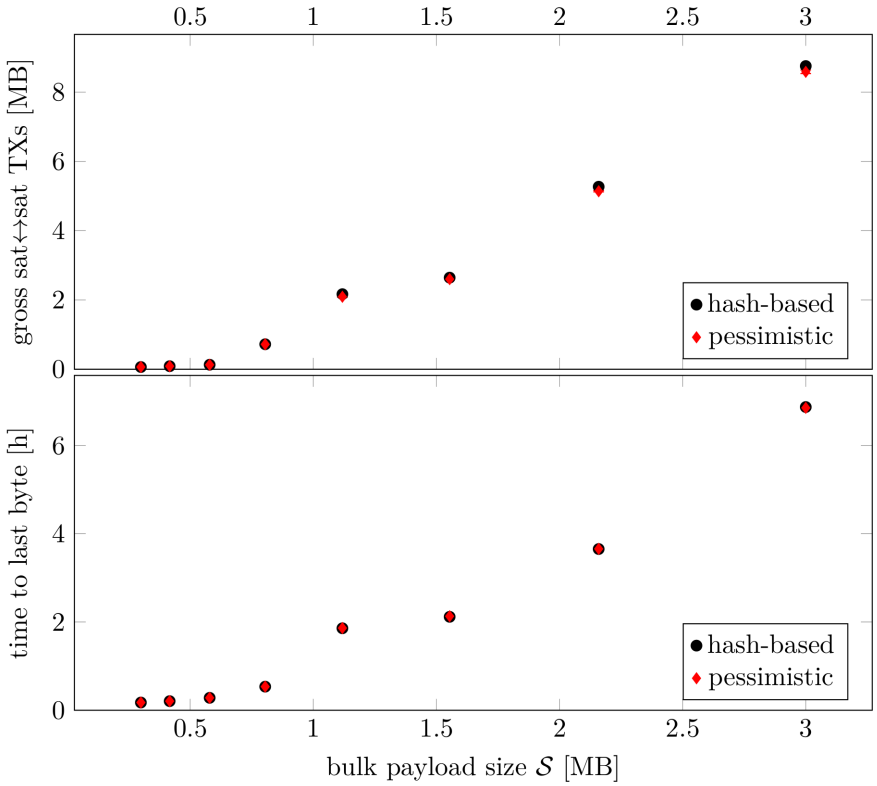


Fig. 7.11.: Time to last byte and gross inter-satellite transmissions depending on bulk payload size for hash-based and pessimistic chunk selection heuristics with generous GS transmission strategy.

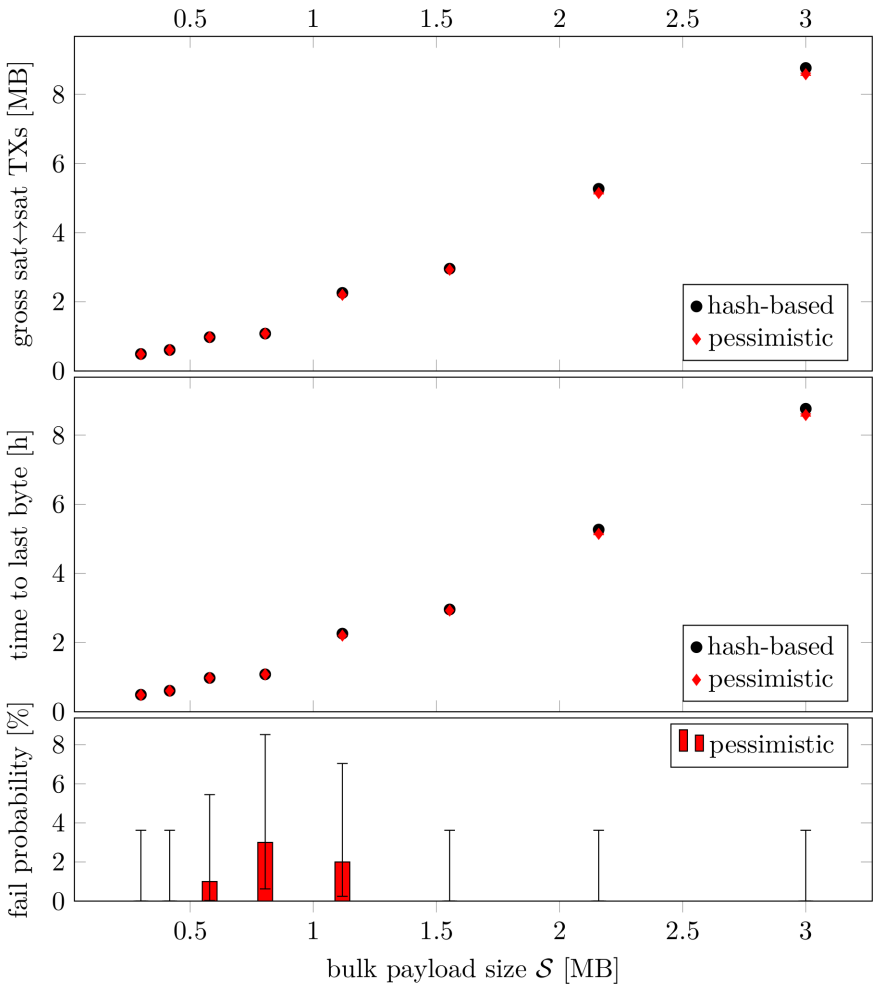


Fig. 7.12.: Time to last byte and gross inter-satellite transmissions depending on bulk payload size for hash-based and pessimistic chunk selection heuristics with frugal GS transmission strategy. The bottom plot shows the probability of pessimistic stall with error bars depicting Clopper–Pearson intervals for 95% confidence level. The corresponding raw measurements have been excluded from the data presented in the upper plots. Each data point was measured with 100 simulation runs. Error bars in the top and middle figures are drawn but are too small to be visible.

effect of a pessimistic stall occurs with a small but measurable probability in case of the pessimistic heuristic. This means that even though the hash-based heuristic does not offer quantitatively measurable benefits over the pessimistic heuristic in those simulation runs where the latter leads to complete transmissions, the pessimistic heuristic leads to an *incomplete protocol* in the sense that, apart from efficiency considerations, the pessimistic-heuristic toy protocol is not even guaranteed to continue service until the whole transmission has been completed. Even though this shortcoming appears to be rather marginal in terms of its fairly small probability, it qualitatively voids the reliability of transmissions.

Just looking at the data presented so far, one could think of solving this issue of the pessimistic heuristic by restricting ground station operation to generous mode. While this would indisputably restore the reliability of transmissions, it still has two disadvantages: first, even in generous mode it can happen that enough data is received in orbit at the end of a contact period, but dissemination using pessimistic heuristic stalls until it resumes with the next contact period. This next contact period could, however, lie many hours in the future, meaning that the pessimistic stall increases the time to last byte enormously. Second, frugal GS mode is actually capable of reducing the consumed GS contact time, as can be seen in Fig. 7.13. In situations where this metric is deemed most important, using generous GS mode is simply suboptimal.

Finally, we compare the toy protocol in both ground station modes to pure single-hop OTAP based on rateless codes, i. e., not using inter-satellite communication at all. Trivially, single-hop OTAP will outperform the toy protocol in terms of minimizing inter-satellite communication demand. Therefore, we measured consumed GS contact period time and time to last byte for the hash-based variant of the toy protocol in frugal and generous ground station mode as well as single-hop OTAP. We did not implement the latter as a usable protocol but conducted simulations only as a best-case estimate: assuming a perfect rateless code, we let the ground station transmit messages using sweep tracking and simply counted the number of successfully received messages at each satellite. Not implementing any feedback mechanism for the single-hop case, we assume that a satellite can decode the bulk payload as soon as $k = \lceil \mathcal{S}/K \rceil$ different messages have been received. In addition, we assume that the GS stops transmitting as soon as all satellites have received the bulk payload. In disregarding both the necessity for a feedback mechanism and the small but finite imperfection of known rateless codes, we try to assure that the results that we measure for our idealized single-hop OTAP protocol are better than the results achievable in practice. The data shown in Fig. 7.13 indicates that the toy protocol significantly outperforms the idealized single-hop OTAP protocol in terms of time-to-last byte when using generous GS mode and in terms of consumed GS contact period time when using frugal mode.

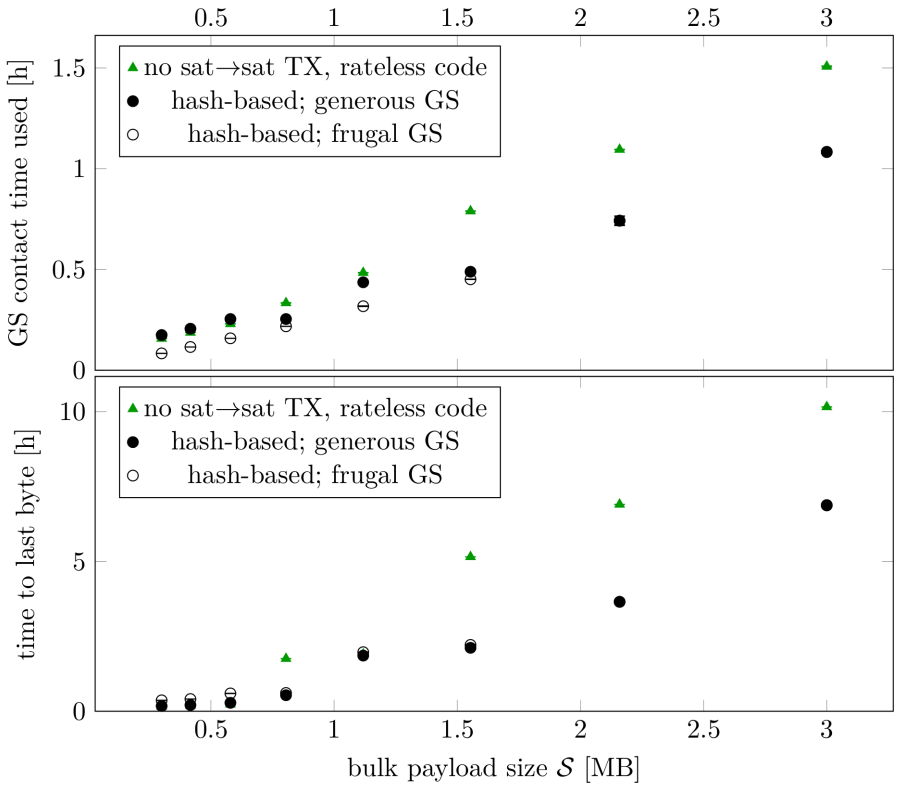


Fig. 7.13.: Comparison of single-hop OTAP using a rateless code to the toy protocol using hash-based chunk selection in generous as well as frugal ground station mode.

7.4 Conclusion

Based on the empirical evaluation of our toy protocol, we have produced two main findings.

First, we have found that a decoder state that uniquely depends on its corresponding linear subspace of injected rows indeed enables, through hashing of decoder state, feedback mechanisms and chunk selection heuristics that increase protocol efficiency. To the best of our knowledge, EAGER decoding is the only known RLNC decoding technique capable of reaching this unique decoder state at low computational overhead after each step of incremental decoding.

Second, the quantitative benefit of using overlapped chunked RLNC over separate, non-overlapping chunks, is rather marginal. It is questionable whether this benefit would even persist in practice, under more realistic channel conditions, a realistic MAC protocol, and interdependence with other satellite-to-satellite communication tasks performed on the same radio channel.

Part III

Epilogue

8 Conclusion

> *ODIN! I await thee*
> *Your true son am I*
> *I hail you*
> *now as I die*
>
> — **Manowar**
> North-American True Metal Band

In Part I we have shown that the exploitation of knowledge that is shared across multiple nodes and the removal of redundancies pay out in terms of protocol efficiency. In the case of the LAMA and CAMELAMA MAC protocols that we presentend in Chapters 2 and 3, the application layer payload, i. e., nodes positions, were used to create a novel robust low-overhead MAC layer mechanism. In doing so we exploited the application layer payload for MAC layer purposes, which means that the LAMA family of protocols can be seen as some esoteric way of cross-layer optimization. In Chapter 4 we demonstrated the benefits of in-orbit forwarding for real-time traffic monitoring. But instead of plain forwarding, we used an in-orbit aggregation method to get rid of outdated payload data. By removing the redundancy that stems from multiple different beacon messages from the same vessel, we reduced communication demand without sacrificing the overall goal of up-to-date navigational states. The strategy of exploiting information redundancy is even clearer in Chapter 5 where we demonstrated the applicability of distributed source coding (DSC) to real-world satellite measurement data. The idea of DSC in turn is, by definition, to use knowledge about redundancies in data from separate data sources to increase coding efficiency and thereby using the available limited channel resources more efficiently. So even though we did not find a one-size-fits-all method to increase the efficiency of payload data retrieval, we have demonstrated for different kinds of links and at different network layers, that it is worth looking for redundancies that could be exploited using purpose-made protocols.

In Part II on the other hand, being much less focussed on *communication protocols*, we have developed a novel decoding technique for RLNC which, due to its feature of bringing decoder states in incremental decoding to a unique representation, enables efficient and robust usage of RLNC for satellite formation OTAP. We have demonstrated the benefits of abandoning the prevailing practice of LU decomposition when solving systems of linear

equations by means of GE. We found that the eager forward substitution and backward substitution do not cause any disadvantages compared to decomposition-based approaches as long as the underlying arithmetic is exact (as opposed to floating point arithmetic). The advantages gained through non-decomposing EAGER decoding, however, are manifold and range from more efficient partial decoding of prioritized layered RLNC over no-overhead joint decodability of chunked RLNC with overlapping chunks to cheaply obtaining a unique decoder state that allows, by means of hash values, a comparison of the vector spaces spanned by equal-rank decoder states. Finally, we have shown that especially the latter turns out useful in the context of satellite OTAP.

Considering the contributions of both parts together, despite being apparently a potpourri of improvements on different layers in somewhat unrelated use cases, we hope to have presented some novel ideas. In the end we have shown in various satellite-formation-related scenarios that even without any ground-breaking innovation, tenacious search for information redundancies and exploitation thereof through adequate communication protocols can significantly pay off in terms of increased efficiency as long as one is willing to hazard the consequences of employing purpose-made protocols.

9 Bibliography

> *Wipe thine ass with What is Written*
> *and grin like a ninny at what is Spoken.*
> *Take thine refuge with thine wine*
> *in the Nothing behind Everything,*
> *as you hurry along the Path.*
>
> — **Malaclypse the Younger**
> from: Principia Discordia

9.0 Peer-Reviewed Co-Authored and Own Publications

[0] Andreas Freimann, Alexander Kleinschrodt, Klaus Schilling, Holger Döbler, and Björn Scheuermann. "Evaluation of a delay tolerant networking approach for inter-satellite communication in LEO for time sensitive traffc monitoring". In: *10th IAA Symposium on Small Satellites for Earth Observation*. 2015 (cit. on pp. 5, 63).

[1] Holger Döbler, Hagen Sparka, and Björn Scheuermann. "Efficient Multi-Satellite Downlinks for Earth Observation Data Based on Distributed Arithmetic Coding". In: *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. 2019, pp. 372–380 (cit. on pp. 6, 75).

[2] Olga Kondrateva, Holger Döbler, Hagen Sparka, et al. "Throughput-Optimal Joint Routing and Scheduling for Low-Earth-Orbit Satellite Networks". In: *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. **(Best Paper Award)**. 2018, pp. 59–66.

[3] H. Döbler and B. Scheuermann. "LAMA: Location-Assisted Medium Access for Position-Beaconing Applications". In: *MSWiM '19: Proceedings of the 22nd ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. Miami Beach, FL, USA, Nov. 2019, pp. 253–260 (cit. on pp. 11, 15, 37, 56, 59, 62).

[4] Holger Döbler and Björn Scheuermann. "CAMELAMA: Cooperative Awareness and space-borne Monitoring Enabled by Location-Assisted Medium Access". In: *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*. **(Best Paper Award)**. 2022, pp. 1–8 (cit. on p. 33).

[5] Holger Döbler and Björn Scheuermann. "CAMELAMA: Cooperative awareness and space-borne monitoring enabled by location-assisted medium access". In: *Computer Communications* (2023). In Press, Journal Pre-proof as of July 2023 (cit. on p. 33).

[6]A. Freimann, T. Petermann, H. Döbler, B. Scheuermann, and K. Schilling. “Efficient Data Uploads to Satellite Formations by Rateless Codes and Adaptive Tracking”. In: *2021 IEEE Space Hardware and Radio Conference (SHaRC)*. 2021, pp. 1–4 (cit. on pp. 131, 135, 149).

9.1 Other Peer-Reviewed Publications

[7]David W Waltrop. “Recovery of the Last GAMBIT and HEXAGON Film Buckets from Space, August–October 1984”. In: *Studies Intel* 58.2 (2014), pp. 19–34 (cit. on p. 7).

[8]Martin N Sweeting. “Space at Surrey: micro-mini-satellites for affordable access to space”. In: *Air & Space Europe* 2.1 (2000), pp. 38–52 (cit. on p. 7).

[9]Hank Heidt, Jordi Puig-Suari, Augustus Moore, Shinichi Nakasuka, and Robert Twiggs. “CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation”. In: (2000) (cit. on p. 7).

[10]Thyrso Villela, Cesar A Costa, Alessandra M Brandão, Fernando T Bueno, and Rodrigo Leonardi. “Towards the thousandth CubeSat: A statistical overview”. In: *International Journal of Aerospace Engineering* 2019 (2019) (cit. on p. 8).

[11]Ricardo Silva, Jorge Sá Silva, and Fernando Boavida. “Mobility in wireless sensor networks–survey and proposal”. In: *Computer Communications* 52 (2014), pp. 1–20 (cit. on p. 8).

[12]Mario Di Francesco, Sajal K Das, and Giuseppe Anastasi. “Data collection in wireless sensor networks with mobile elements: A survey”. In: *ACM Transactions on Sensor Networks (TOSN)* 8.1 (2011), pp. 1–31 (cit. on p. 9).

[13]Felicia Engmann, Ferdinand Apietu Katsriku, Jamal-Deen Abdulai, Kofi Sarpong Adu-Manu, and Frank Kataka Banaseka. “Prolonging the lifetime of wireless sensor networks: a review of current techniques”. In: *Wireless Communications and Mobile Computing* 2018 (2018) (cit. on p. 9).

[14]Charles Toth and Grzegorz Jóźków. “Remote sensing platforms and sensors: A survey”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 115 (2016), pp. 22–36 (cit. on p. 9).

[15]F. F. Mobley, L. D. Eckard, G. H. Fountain, and G. W. Ousley. “Magsat - A new satellite to survey the earth’s magnetic field”. In: *IEEE Transactions on Magnetics* 16 (Sept. 1980), pp. 758–760 (cit. on pp. 9, 76).

[16]A. Rajandekar and B. Sikdar. “A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications”. In: *IEEE Internet of Things Journal* 2 (2 2015), pp. 175–186 (cit. on pp. 16, 17).

[17]Z. Haas and J. Deng. “Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks”. In: *IEEE Transactions on Communications* 50 (6 2002), pp. 975–985 (cit. on p. 17).

[18]Floriano De Rango, Annalisa Perrotta, and Saverio Ombres. “A energy evaluation of E-TDMA vs IEEE 802.11 in wireless ad hoc networks”. In: *Proceedings of the 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS ’10)*. 2010, pp. 273–279 (cit. on p. 17).

[19]V. Toldov, L. Clavier, and N. Mitton. "Multi-channel Distributed MAC protocol for WSN-based wildlife monitoring". In: *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2018, pp. 1–8 (cit. on p. 17).

[20]M. Salajegheh, H. Soroush, and A. Kalis. "HYMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks". In: *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. 2007, pp. 1–5 (cit. on p. 17).

[21]C. Young. "USAP: a unifying dynamic distributed multichannel TDMA slot assignment protocol". In: *Proceedings of MILCOM '96 IEEE Military Communications Conference*. 1996, 235–239–vol.1 (cit. on p. 17).

[22]X. Jiang and D. Du. "PTMAC: A Prediction-Based TDMA MAC Protocol for Reducing Packet Collisions in VANET". In: *IEEE Transactions on Vehicular Technology* 65 (11 2016), pp. 9209–9223 (cit. on p. 17).

[23]F. Klingler, R. Cohen, C. Sommer, and F. Dressler. "Bloom Hopping: Bloom Filter Based 2-Hop Neighbor Management in VANETs". In: *IEEE Transactions on Mobile Computing* 18 (3 2019), pp. 534–545 (cit. on p. 17).

[24]C. Zhu and M. Corson. "A Five-Phase Reservation Protocol (FPRP) for mobile Ad Hoc networks". In: *Wireless Networks* 7 (4 2001), pp. 371–384 (cit. on p. 17).

[25]A. Bazzi, B. Masini, A. Zanella, and I. Thibault. "On the Performance of IEEE 802.11p and LTE-V2V for the Cooperative Awareness of Connected Vehicles". In: *IEEE Transactions on Vehicular Technology* 66 (11 2017), pp. 10419–10432 (cit. on p. 18).

[26]K. Bilstrup, E. Uhlemann, E. Strom, and U. Bilstrup. "Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication". In: *2008 IEEE 68th Vehicular Technology Conference*. 2008, pp. 1–5 (cit. on p. 18).

[27]H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein. "Contention-based forwarding for mobile ad hoc networks". In: *Ad Hoc Networks* 1 (4 2003), pp. 351–369 (cit. on p. 18).

[28]J. Pathmasuntharam, J. Jurianto, P. Kong, et al. "High Speed Maritime Ship-to-Ship/Shore Mesh Networks". In: *2007 7th International Conference on ITS Telecommunications*. 2007, pp. 1–6 (cit. on p. 24).

[29]I. Timmins and S. O'Young. "Marine Communications Channel Modeling Using the Finite-Difference Time Domain Method". In: *IEEE Transactions on Vehicular Technology* 58 (6 2009), pp. 2626–2637 (cit. on p. 24).

[30]S. Gowrishankar, T. Basavaraju, and Subir Sarkar. "Effect of random mobility models pattern in mobile ad hoc networks". In: *IJCSNS* 7.6 (2007), pp. 160–164 (cit. on p. 25).

[31]Michael Wing, Aaron Eklund, and Loren Kellogg. "Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability". In: *Journal of Forestry* 103.4 (June 1, 2005), pp. 169–173 (cit. on p. 30).

[32]Nathan G Orr, Jeff Cain, Luke Stras, and Robert E Zee. "Space based AIS detection with the maritime monitoring and messaging microsatellite". In: *64th International Astronautical Congress*. 2013 (cit. on p. 34).

[33]Jesper A Larsen and Hans Peter Mortensen. “In orbit validation of the AAUSAT3 SDR based AIS receiver”. In: *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*. IEEE. 2013, pp. 487–491 (cit. on p. 34).

[34]D. Kocak and P. Browning. “Real-time AIS tracking from space expands opportunities for global ocean observing and maritime domain awareness”. In: *OCEANS’15*. Washington, DC, USA, Oct. 2015, pp. 1–7 (cit. on pp. 34, 36, 54).

[35]C. Eisler, P. Dobias, and K. Macneil. “A surveillance application of satellite AIS utilizing a parametric model for probability of detection”. In: *ICORES ’17:Proceedings of the 6th International Conference on Operations Research and Enterprise Systems*. Porto, Portugal, Feb. 2017, pp. 211–218 (cit. on pp. 34, 36, 54).

[36]S. Jayasimha, J. Paladugula, A. Gadiraju, and M. Medam. “Satellite-based AIS receiver for dense maritime zones”. In: *COMSNETS ’17: Proceedings of the 9th International Conference on Communication Systems and Networks*. Bengaluru, India, Jan. 2017, pp. 15–22 (cit. on pp. 34, 36, 62).

[37]A. Skauen. “Ship tracking results from state-of-the-art space-based AIS receiver systems for maritime surveillance”. In: *CEAS Space Journal* 11 (3 2019), pp. 301–316 (cit. on p. 36).

[38]W. Hasbi, K. Mukhayadi, and U. Renner. “The impact of space-based AIS antenna orientation on in-orbit AIS detection performance”. In: *Applied Sciences (Switzerland)* 9 (16 2019) (cit. on p. 36).

[39]S. Li, X. Chen, L. Chen, et al. “Data reception analysis of the AIS on board the TianTuo-3 satellite”. In: *Journal of Navigation* 70 (4 2017), pp. 761–774 (cit. on p. 36).

[40]F. Clazzer, F. Lázaro, and S. Plass. “Enhanced AIS receiver design for satellite reception”. In: *CEAS Space Journal* 8 (4 2016), pp. 257–268 (cit. on p. 36).

[41]F. Maggio, T. Rossi, E. Cianca, and M. Ruggieri. “Digital beamforming techniques applied to satellite-based AIS receiver”. In: *IEEE Aerospace and Electronic Systems Magazine* 29 (6 2014), pp. 4–12 (cit. on pp. 36, 62).

[42]A. Harchowdhury, B. Sarkar, K. Bandyopadhyay, and A. Bhattacharya. “Generalized mechanism of SOTDMA and probability of reception for satellite-based AIS”. In: *CODEC ’12: Proceedings of the 5th International Conference on Computers and Devices for Communication*. Kolkata, India, Dec. 2012, pp. 1–4 (cit. on p. 36).

[43]E. Casini, R. De Gaudenzi, and O. Del Rio Herrero. “Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks”. In: *IEEE Transactions on Wireless Communications* 6 (4 2007), pp. 1408–1419 (cit. on p. 36).

[44]S. Gollakota and D. Katabi. “Zigzag decoding: combating hidden terminals in wireless networks”. In: 38 (4 2008), pp. 159–170 (cit. on p. 36).

[45]E. Paolini, C. Stefanovic, G. Liva, and P. Popovski. “Coded random access: applying codes on graphs to design random access protocols”. In: *IEEE Communications Magazine* 53 (6 2015), pp. 144–150 (cit. on p. 36).

[46]S. Akin and M. Fidler. “Multi-Access Spreading over Time: MAST”. In: *MSWiM ’19: Proceedings of the 22nd ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. Miami Beach, FL, USA, Nov. 2019, pp. 261–270 (cit. on p. 36).

[47]M. Shih, G. Lin, and H. Wei. “A Distributed Multi-Channel Feedbackless MAC Protocol for D2D Broadcast Communications”. In: *IEEE Wireless Communications Letters* 4 (1 2015), pp. 102–105 (cit. on p. 36).

[48]J. Mao, S. Chen, Y. Liu, J. Yu, and Y. Xu. “LT-MAC: A location-based TDMA MAC protocol for small-scale underwater sensor networks”. In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2015, pp. 1275–1280 (cit. on p. 36).

[49]H. Jang, E. Kim, J. Lee, and J. Lim. “Location-Based TDMA MAC for Reliable Aeronautical Communications”. In: *IEEE Transactions on Aerospace and Electronic Systems* 48 (2 2012), pp. 1848–1854 (cit. on p. 36).

[50]T. Tarnai and Z. Gáspár. “Multi-symmetric close packings of equal spheres on the spherical surface”. In: *Acta Crystallographica Section A* 43 (5 1987), pp. 612–616 (cit. on pp. 39, 45).

[51]W. Ridgway and A. Cheviakov. “An iterative procedure for finding locally and globally optimal arrangements of particles on the unit sphere”. In: *Computer Physics Communications* 233 (2018), pp. 84–109 (cit. on pp. 39, 40).

[52]L. Toth. “Über die dichteste Kugellagerung”. In: *Mathematische Zeitschrift* 48 (1 1942), pp. 676–684 (cit. on p. 45).

[53]R. Robinson. “Arrangement of 24 points on a sphere”. In: *Mathematische Annalen* 144 (1 1961), pp. 17–48 (cit. on p. 45).

[54]E. Ben Hamida, G. Chelius, and M. Gorce. “Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation”. In: *Simulation* 85 (9 2009), pp. 574–588 (cit. on p. 54).

[55]J. Yoon, M. Liu, and B. Noble. “Random waypoint considered harmful”. In: *INFOCOM ’03: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*. San Francisco, CA, USA, Mar. 2003, 1312–1321–vol.2 (cit. on p. 56).

[56]A. Freimann, M. Dierkes, T. Petermann, et al. “ESTNeT: a discrete event simulator for space-terrestrial networks”. In: *CEAS Space Journal* 13 (1 2021), pp. 39–49 (cit. on pp. 56, 148).

[57]Philip Bangert, Stephan Busch, and Klaus Schilling. “Performance characteristics of the UWE-3 miniature attitude determination and control system”. In: *2nd IAA Conference on Dynamics and Control of Space Systems (DYCOSS)*. 2014 (cit. on p. 66).

[58]David S. Slepian and Jack K. Wolf. “Noiseless coding of correlated information sources”. In: *IEEE Trans. Information Theory* 19.4 (1973), pp. 471–480 (cit. on pp. 75, 77, 79).

[59]Marco Grangetto, Enrico Magli, and Gabriella Olmo. “Distributed Arithmetic Coding”. In: *IEEE Communications Letters* 11.11 (2007), pp. 883–885 (cit. on pp. 76, 79–82, 84).

[60]Naoto Kimura and Shahram Latifi. “A Survey on Data Compression in Wireless Sensor Networks”. In: *ITCC (2)*. IEEE Computer Society, 2005, pp. 8–13 (cit. on pp. 76, 81).

[61]Mohammad Abdur Razzaque, Chris J. Bleakley, and Simon Dobson. “Compression in wireless sensor networks: A survey and comparative evaluation”. In: *TOSN* 10.1 (2013), 5:1–5:44 (cit. on p. 77).

[62]Kenneth C. Barr and Krste Asanovic. “Energy-aware lossless data compression”. In: *ACM Trans. Comput. Syst.* 24.3 (2006), pp. 250–291 (cit. on pp. 77, 87).

[63]S. Sandeep Pradhan and Kannan Ramchandran. “Distributed Source Coding Using Syndromes (DISCUS): Design and Construction”. In: *Data Compression Conference*. IEEE Computer Society, 1999, pp. 158–167 (cit. on p. 77).

[64]Javier Garcia-Frias. “Joint Source-Channel Decoding of Correlated Sources over Noisy Channels”. In: *Data Compression Conference*. IEEE Computer Society, 2001, pp. 283–292 (cit. on p. 77).

[65]Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades. “Joint source-channel coding of binary sources with side information at the decoder using IRA codes”. In: *IEEE Workshop on Multimedia Signal Processing*. IEEE, 2002, pp. 53–56 (cit. on p. 77).

[66]Marco Grangetto, Enrico Magli, and Gabriella Olmo. “Distributed arithmetic coding for the Slepian-Wolf problem”. In: *IEEE Trans. Signal Processing* 57.6 (2009), pp. 2245–2257 (cit. on pp. 77, 82).

[67]Ian H. Witten, Radford M. Neal, and John G. Cleary. “Arithmetic Coding for Data Compression”. In: *Commun. ACM* 30.6 (1987), pp. 520–540 (cit. on pp. 77, 78, 80).

[68]Xi Chen and David S. Taubman. “Coupled distributed arithmetic coding”. In: *ICIP*. IEEE, 2011, pp. 341–344 (cit. on pp. 77, 82).

[69]J. Zhou, K. W. Wong, and J. Chen. “Distributed Block Arithmetic Coding for Equiprobable Sources”. In: *IEEE Sensors Journal* 13.7 (July 2013), pp. 2750–2756 (cit. on p. 77).

[70]Li Chen and David S. Taubman. “Distributed source coding based on punctured conditional arithmetic codes”. In: *ICIP*. IEEE, 2010, pp. 3713–3716 (cit. on p. 77).

[71]Yong Fang. “DAC Spectrum of Binary Sources with Equally-Likely Symbols”. In: *IEEE Trans. Communications* 61.4 (2013), pp. 1584–1594 (cit. on p. 77).

[72]Marco Grangetto, Enrico Magli, and Gabriella Olmo. “Distributed joint source-channel arithmetic coding”. In: *ICIP*. IEEE, 2010, pp. 3717–3720 (cit. on pp. 77, 82).

[73]J. Wu, M. Wang, J. Jeong, and L. Jiao. “Adaptive-distributed arithmetic coding for lossless compression”. In: *2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content*. Sept. 2010, pp. 541–545 (cit. on p. 77).

[74]Yong Fang. “Distribution of Distributed Arithmetic Codewords for Equiprobable Binary Sources”. In: *IEEE Signal Process. Lett.* 16.12 (2009), pp. 1079–1082 (cit. on p. 77).

[75]Yasaman Keshtkarjahromi, Mehrdad Valipour, and Farshad Lahouti. “Multi-level Distributed Arithmetic Coding with Nested Lattice Quantization”. In: *DCC*. IEEE, 2014, pp. 382–391 (cit. on pp. 77, 82).

[76]Xavier Artigas, Simon Malinowski, Christine Guillemot, and Luis Torres. “Overlapped arithmetic codes with memory”. In: *EUSIPCO*. IEEE, 2008, pp. 1–5 (cit. on p. 77).

[77]Nikos Deligiannis, Evangelos Zimos, Dragos Mihai Ofrim, Yiannis Andreopoulos, and Adrian Munteanu. “Distributed Joint Source-Channel Coding with Raptor Codes for Correlated Data Gathering in Wireless Sensor Networks”. In: *BODYNETS*. ICST, 2014 (cit. on p. 77).

[78]D. Wang and L. Huang. "HMM based distributed arithmetic coding and its application in image coding". In: *Fifth International Conference on Machine Vision (ICMV 2012): Computer Vision, Image Analysis and Processing*. Vol. 8783. Mar. 2013 (cit. on p. 77).

[79]A. Abrardo, M. Barni, E. Magli, and F. Nencini. "Error-Resilient and Low-Complexity Onboard Lossless Compression of Hyperspectral Images by Means of Distributed Source Coding". In: *IEEE Transactions on Geoscience and Remote Sensing* 48.4 (Apr. 2010), pp. 1892–1904 (cit. on p. 78).

[80]D. A. Huffman. "A Method for the Construction of Minimum-Redundancy Codes". In: *Proceedings of the IRE* 40 (1952) (cit. on p. 78).

[81]Claude Elwood Shannon. "A mathematical theory of communication". In: *Bell system technical journal* 27.3 (1948), pp. 379–423 (cit. on p. 78).

[82]Alistair Moffat, Radford M. Neal, and Ian H. Witten. "Arithmetic Coding Revisited". In: *ACM Trans. Inf. Syst.* 16.3 (1998), pp. 256–294 (cit. on p. 80).

[83]A. Moffat. "Implementing the PPM data compression scheme". In: *IEEE Transactions on Communications* 38.11 (Nov. 1990), pp. 1917–1921 (cit. on p. 81).

[84]I. H. Witten and T. C. Bell. "The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression". In: *IEEE Transactions on Information Theory* 37.4 (1991), pp. 1085–1094 (cit. on p. 81).

[85]Marco Grangetto, Pamela C. Cosman, and Gabriella Olmo. "Joint source/channel coding and MAP decoding of arithmetic codes". In: *IEEE Trans. Communications* 53.6 (2005), pp. 1007–1016 (cit. on pp. 82, 84).

[86]Junwei Zhou, Kwok-Wo Wong, and Yanchao Yang. "Distributed arithmetic coding with interval swapping". In: *Signal Processing* 116 (2015), pp. 29–37 (cit. on p. 82).

[87]Bryan Klofas and Kyle Leveque. "The future of CubeSat communications: transitioning away from amateur radio frequencies for high-speed downlinks". In: *Proceedings of AMSAT space symposium*. 2012 (cit. on p. 87).

[88]R. Naumann, S. Dietzel, and B. Scheuermann. "Best of Both Worlds: Prioritizing Network Coding without Increased Space Complexity". In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. 2016, pp. 723–731 (cit. on pp. 96, 97, 103, 115, 116).

[89]M. Luby. "LT codes". In: *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* 2002, pp. 271–280 (cit. on p. 97).

[90]A. Shokrollahi. "Raptor codes". In: *IEEE Transactions on Information Theory* 52 (6 2006), pp. 2551–2567 (cit. on p. 97).

[91]J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. "A digital fountain approach to reliable distribution of bulk data". In: *Computer Communication Review* 28 (4 1998), pp. 56–67 (cit. on p. 97).

[92]M. Champel, K. Huguenin, A. Kermarrec, and N. Scouarnec. "LT Network Codes". In: *2010 IEEE 30th International Conference on Distributed Computing Systems*. 2010, pp. 536–546 (cit. on p. 97).

[93]J. Heide, M. Pedersen, F. Fitzek, and M. Medard. "A Perpetual Code for Network Coding". In: *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*. 2014, pp. 1–6 (cit. on pp. 97, 105).

[94]A. Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, and E. Magli. "Band Codes for Energy-Efficient Network Coding With Application to P2P Mobile Streaming". In: *IEEE Transactions on Multimedia* 16 (2 2014), pp. 521–532 (cit. on pp. 97, 104, 105).

[95]Y. Li, W. Chan, and S. Blostein. "On Design and Efficient Decoding of Sparse Random Linear Network Codes". In: *IEEE Access* 5 (2017), pp. 17031–17044 (cit. on pp. 97, 100).

[96]Y. Li, J. Zhu, and Z. Bao. "Sparse Random Linear Network Coding With Precoded Band Codes". In: *IEEE Communications Letters* 21 (3 2017), pp. 480–483 (cit. on p. 97).

[97]D. Silva, W. Zeng, and F. Kschischang. "Sparse network coding with overlapping classes". In: *2009 Workshop on Network Coding, Theory, and Applications*. 2009, pp. 74–79 (cit. on pp. 97, 100, 133).

[98]R. Ngeth, B. Kurkoski, Y. Lim, and Y. Tan. "A design of overlapped chunked code over compute-and-forward for multi-source multi-relay networks". In: *Sensors (Switzerland)* 18 (10 2018) (cit. on pp. 97, 133).

[99]S. Alam, S. Sultana, Y. Hu, and S. Fahmy. "SYREN: Synergistic Link Correlation-Aware and Network Coding-Based Dissemination in Wireless Sensor Networks". In: *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*. 2013, pp. 485–494 (cit. on pp. 98, 133, 135).

[100]A. Hagedorn, D. Starobinski, and A. Trachtenberg. "Rateless Deluge: Over-the-Air Programming of Wireless Sensor Networks Using Random Linear Codes". In: *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*. 2008, pp. 457–466 (cit. on pp. 98, 133, 135).

[101]K. Nguyen, T. Nguyen, and S. Cheung. "Peer-to-Peer Streaming with Hierarchical Network Coding". In: *2007 IEEE International Conference on Multimedia and Expo*. 2007, pp. 396–399 (cit. on pp. 101, 115).

[102]C. Fragouli, J. Le Boudec, and J. Widmer. "Network coding: an instant primer". In: *Computer Communication Review* 36 (1 2006), pp. 63–68 (cit. on pp. 103, 104, 133).

[103]A. Heidarzadeh and A. Banihashemi. "Overlapped Chunked network coding". In: *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*. 2010, pp. 1–5 (cit. on pp. 121, 122, 133).

[104]C. Studholme and I. Blake. "Random Matrices and Codes for the Erasure Channel". In: *Algorithmica (New York)* 56 (4 2010), pp. 605–620 (cit. on p. 122).

[105]Klaus Schilling, Tristan Tzschichholz, Iurii Motroniuk, et al. "TOM: A Formation for Photogrammetric Earth Observation by Three CubeSats". In: *4th IAA Conference on University Satellite Missions* (Roma, Italy). 2017 (cit. on p. 129).

[106]M. Tzabari, V. Holodovsky, O. Shubi, et al. "CloudCT 3D volumetric tomography: considerations for imager preference, comparing visible light, short-wave infrared, and polarized imagers". In: 2021 (cit. on pp. 129, 150).

[107]S. Busch, P. Bangert, S. Dombrovski, and K. Schilling. “UWE-3, in-orbit performance and lessons learned of a modular and flexible satellite bus for future pico-satellite formations”. In: *Acta Astronautica* 117 (2015), pp. 73–89 (cit. on pp. 130, 148).

[108]J. Hui and D. Culler. “The dynamic behavior of a data dissemination protocol for network programming at scale”. In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. Baltimore, MD, USA, 2004, pp. 81–94 (cit. on pp. 133, 134).

[109]J. Subramanian, R. Morris, and H. Balakrishnan. “UFlood: High-throughput flooding over wireless mesh networks”. In: *2012 Proceedings IEEE INFOCOM*. 2012, pp. 82–90 (cit. on pp. 133, 134).

[110]M. Doddavenkatappa, M. Chan, and B. Leong. “Splash: fast data dissemination with constructive interference in wireless sensor networks”. In: *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*. Lombard, IL, 2013, pp. 269–282 (cit. on pp. 133, 134).

[111]V. Naik, A. Arora, P. Sinha, and H. Zhang. “Sprinkler: a reliable and energy efficient data dissemination service for wireless embedded devices”. In: *26th IEEE International Real-Time Systems Symposium (RTSS’05)*. 2005, 10–pp.–286 (cit. on p. 134).

[112]L. Huang and S. Setia. “CORD: Energy-Efficient Reliable Bulk Data Dissemination in Sensor Networks”. In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. 2008, pp. 574–582 (cit. on p. 134).

[113]Z. Zhao, Z. Wang, G. Min, and Y. Cao. “Highly-Efficient Bulk Data Transfer for Structured Dissemination in Wireless Embedded Network Systems”. In: *Journal of Systems Architecture* 72 (2017), pp. 19–28 (cit. on p. 134).

[114]W. Dong, C. Chen, X. Liu, J. Bu, and Y. Gao. “A Lightweight and Density-Aware Reprogramming Protocol for Wireless Sensor Networks”. In: *IEEE Transactions on Mobile Computing* 10 (10 2011), pp. 1403–1415 (cit. on p. 134).

[115]S. Katti, H. Rahul, W. Hu, et al. “XORs in the air: practical wireless network coding”. In: 36 (4 2006), pp. 243–254 (cit. on p. 135).

[116]I. Hou, Y. Tsai, T. Abdelzaher, and I. Gupta. “AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks”. In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. 2008, pp. 1517–1525 (cit. on p. 135).

[117]M. Rossi, N. Bui, G. Zanca, et al. “SYNAPSE++: Code Dissemination in Wireless Sensor Networks Using Fountain Codes”. In: *IEEE Transactions on Mobile Computing* 9 (12 2010), pp. 1749–1765 (cit. on p. 135).

[118]N. Dos Santos Ribeiro Jr., M. Vieira, L. Vieira, and O. Gnawali. “CodeDrip: Data dissemination protocol with network coding for wireless sensor networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8354 LNCS (2014), pp. 34–49 (cit. on p. 135).

9.2 Other Resources

[119]URL: https://xkcd.com/1343/ (visited on Feb. 22, 2022) (cit. on p. 3).

[120]URL: https://www.gnu.org/software/make/manual/html_node/Reading.html (visited on Feb. 22, 2022) (cit. on p. 3).

[121]E. Dijkstra. *Why numbering should start at zero*. Manuscript EWD 831. Aug. 1982 (cit. on p. 4).

[122]Roger D Lanius, John M Logsdon, and Robert W Smith. *Reconsidering Sputnik: Forty years since the Soviet satellite*. Routledge, 2013 (cit. on p. 7).

[123]European Telecommunications Standards Institute. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Co-operative Awareness Basic Service*. Apr. 2010 (cit. on p. 15).

[124]*IEEE P1609.4 standard for wireless access in vehicular environments (WAVE)-multi-channel operation. draft standard ed.* 2006 (cit. on p. 15).

[125]Federal Aviation Administration. *Automatic Dependent Surveillance—Broadcast (ADS–B) Out Performance Requirements To Support Air Traffic Control (ATC) Service; Final Rule*. 14 CFR Part 91. May 28, 2010 (cit. on p. 15).

[126]*Recommendation ITU-R M.1371-5: Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band*. Tech. rep. Feb. 2014 (cit. on pp. 15, 17, 24, 36).

[127]Chenxi Zhu and M. Corson. *An evolutionary-TDMA scheduling protocol (E-TDMA) for mobile ad hoc networks*. University of Maryland, 1998 (cit. on p. 17).

[128]G. Riley and T. Henderson. *The ns-3 network simulator*. 2010, pp. 15–34 (cit. on pp. 24, 54).

[129]URL: ftp://ftp.ais.dk/ais_data/dk_csv_jun2017.rar (visited on Dec. 2, 2018) (cit. on p. 25).

[130]G. Høye. *Ship detection probability analysis for a possible long-range AIS system*. Tech. rep. FFI/RAPPORT-2004/04383. Forsvarets forskningsinstitutt, 2004 (cit. on pp. 36, 68).

[131]R. Roy. "Random Walk Mobility". In: *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer US, 2011 (cit. on p. 56).

[132]G. Eiden and R. Goldsmith. *Performance of AIS sensors in space - PASTA-MARE project final report executive summary*. Tech. rep. European Commission Maritime Forum, 2010 (cit. on p. 66).

[133]The CubeSat Program. *CubeSat Design Specification*. 2014. URL: http://www.cubesat.org/s/cds_rev13_final2.pdf (visited on June 27, 2017) (cit. on p. 75).

[134]Paul G. Howard and Jeffrey Scott Vitter. "Practical Implementations of Arithmetic Coding". In: *Image and Text Compression*. Ed. by James A. Storer. Boston, MA: Springer US, 1992, pp. 85–112 (cit. on pp. 77, 81, 84).

[135]Thomas Boutell. *PNG (portable network graphics) specification version 1.0*. Tech. rep. 1997 (cit. on p. 78).

[136]Amir Said. “Chapter 5 - Arithmetic Coding”. In: *Lossless Compression Handbook*. Ed. by Khalid Sayood. Communications, Networking and Multimedia. San Diego: Academic Press, 2003, pp. 101–152 (cit. on p. 80).

[137]National Aeronautics and Space Administration. *MagSat*. 2017. URL: `https://nssdc.gsfc.nasa.gov/nmc/spacecraftOrbit.do?id=1979-094A` (visited on July 17, 2017) (cit. on p. 87).

[138]URL: `http://www.netlib.org/lapack/lug/node26.html` (visited on Feb. 22, 2022) (cit. on p. 102).

[139]Howard Anton and Anke Walz. *Lineare Algebra: Einführung, Grundlagen, Übungen*. Spektrum Akademischer Verlag, 1998 (cit. on p. 113).

[140]URL: `https://www.burtleburtle.net/bob/hash/spooky.html` (visited on May 23, 2022) (cit. on p. 126).

[141]Miami Beach, FL, USA, Nov. 2019.

Appendix

A

A.0 Glossary

AC arithmetic coding. 80, 81, 87, 91, 92

ADS--B automatic dependent surveillance–broadcast. 7, 13, 31, 63

AIS automatic identification system. 3, 7, 8, 13, 14, 16, 22, 23, 29, 31, 33, 34, 50, 51, 55, 59–61, 63–65, 67, 68, 184

BS base station. 15

BSREF bilateral spread row echelon form. 107, 109, 111, 114, 116, 126–128

CAMELAMA Cooperative Awareness and spaceborne Monitoring Enabled by Location-Assisted Medium Access. 29, 32, 33, 35

CI confidence interval. 24, 118, 119

COTS Commercial-Off-The-Shelf. 5

CSMA carrier sense multiple access. 144

DAC distributed arithmetic coding. xi, 4, 76–85, 87, 88, 90–92

DES discrete event simulation. 33, 51

DOS denial-of-service. 48

DSC distributed source coding. 8, 77, 78, 171

ECEF Earth-centered, Earth-fixed. 55

EV encoding vector. 99, 100, 105, 106, 111, 112, 114, 116, 119, 121, 122

FoV field of view. 34, 56, 59

FPRP Five Phase Reservation Protocol. 15

GE Gaussian elimination. xi, 95, 96, 102–104, 111, 112, 116, 171

GNSS global navigation satellite system. 9

ground station A satellite ground station is a terrestrial node that is involved in the operation of one or more satellites. Terrestrial nodes not involved in missions operation, such as vessels broadcasting automatic identification system (AIS) beacons, are excluded from this definition despite their communication to satellites. 7

GS ground station. 9, 63, 68, 75, 131, 132, 134–142, 147, 151–154, 156–159, 161–167

HNC Hierarchical Network Coding. 112, 116–118

HPBW half-power beam width. 55, 56, 58, 59

HTP hidden terminal problem. 14, 26, 28, 34, 142, 144

IOA in-order assumption. 117

ISL inter-satellite link. 8, 63

IV information vector. 97, 99, 100, 105, 106

LAMA location-assisted medium access for beaconing applications. 8, 26, 29, 32, 33, 35

LC linear combination. 95–97, 99, 100, 102–106, 109, 111, 112, 114–117, 119–125, 127, 129, 136, 140, 141, 144, 147–150, 160, 162

LEO low Earth orbit. 5, 7, 8, 31–35, 49, 50, 129, 134, 137

MAC Medium Access Control. 3, 8, 13, 14, 16, 22, 29, 31–35, 55, 60, 135, 140, 142, 160, 168, 171

MANET Mobile Ad-hoc Network. 6

OTAP Over-the-Air Programming. 9, 129, 131, 134–139, 141, 151, 152, 154, 155, 162, 166–168, 171, 172

PER packet error rate. 23

RAAN right ascension of the ascending node. 152

REF row echelon form. 107

RLNC random linear network coding. 9, 95–97, 99, 103–105, 117–119, 124, 127–129, 132, 134, 135, 138, 139, 142, 151, 152, 154, 157, 160, 162, 167, 168, 171, 172

RSSI received signal strength indicator. 48

RTT round-trip time. 143

S-AIS Satellite-AIS. 8, 31, 33

SGE Swap Gaussian Elimination. 109, 111

SIC successive interference cancellation. 33, 34

Slerp spherical linear interpolation. 51

SO-TDMA self-organizing time division multiple access. 3, 14–16, 22, 24, 26, 28, 29, 31, 33, 34, 50, 52, 55, 56, 58–61, 64, 67, 68

SREF spread row echelon form. 107

surface The Earth's surface; all nodes in the Earth's atmosphere, resting or moving at suborbital speed, are termed "surface nodes". 185

TDMA time division multiple access. 15, 16, 140, 144

terrestrial node Synonym for surface node. 184

UAR uniformly at random. 51, 54, 141

USAP Unifying Dynamic Distributed Multichannel TDMA Slot Assignment protocol. 15

WSN wireless sensor network. 6, 7, 9, 131, 134, 136, 137

ZfT Zentrum für Telematik e.V.. 131

A.1 Glosarry of Mathematical Notation

$[a, b)_{\mathbb{Z}}$	$:= \{i \in \mathbb{Z} \colon a \leq i < b\}$, integer interval not containing its nominal upper bound.
$[a, b]_{\mathbb{Z}}$	$:= \{i \in \mathbb{Z} \colon a \leq i \leq b\}$, integer interval containing its nominal upper bound.
$\mathrm{frac}(x)$	$:= x - \lfloor x \rfloor \; \forall x \in \mathbb{R}$, the factional part of a real number.
$\max_f S$	Maximum of set S with respect to function f, i. e., $\max_f S := \operatorname{argmax}_{x \in S} f(x)$. Whether we mean the unique argmax element, any of the argmax elements or the whole argmax set depends on the context.
$\min_f S$	See $\max_f S$.
$\mathrm{Slerp}(p_0, p_1; t)$	$:= \frac{\sin((1-t)\Omega)}{\sin \Omega} p_0 + \frac{\sin(t\Omega)}{\sin \Omega} p_1$, where Ω is the angle between the vectors p_0 and p_1, denotes spherical linear interpolation.
$\lvert A \rvert$	where A is a set denotes the cardinality of A.
$\lvert a \rvert$	where a is a real or complex number denotes the absolute value of a.
$\lVert \mathbf{a} \rVert$	where $\mathbf{a}$ is an element of a normed vector space denotes the norm of $\mathbf{a}$. If not stated differently, we assume the euclidean norm.
$\mathcal{P}S$	is the powerset of a set S.

EAGER notation:

0 and 1 Talking about elements of $\mathbb{F}_q$, we write 0 for the additive identity and 1 for the multiplicative identity of the finite field.

eliminate By "eliminate a using b" we mean, that given two vectors $a, b \in \mathbb{F}_q^n$ with $h(a) = h(b) =: i$, we replace a with $a - \frac{a_i}{b_i} b$, leaving b unaltered.

back-end eliminate By "back-end eliminate a using b" mean, that given two vectors $a, b \in \mathbb{F}_q^n$ with $t(a) = t(b) =: i$, we replace a with $a - \frac{a_i}{b_i} b$, leaving b unaltered.

front-end eliminate Same as "eliminate".

A, B, G, H, L, U, X, Y Capital letters denote matrices. B is sometimes used for a basis (tuple of linearly independent vectors).

$\operatorname{span} A$ The linear span of the rows of a matrix A.

A_i The i^{th} row of a given matrix, i. e., a vector.

A_{ij} The j^{th} element of A_i.

$U^{(i)}$ denotes the i^{th} element of a sequence of matrices, where it is assumed that the computation of $U^{(i+1)}$ based on $U^{(i)}$ can be implemented in-place.

$a \equiv 0$ All elements of a are zero. When referring to a matrix row, e. g., $G_i \equiv 0$, this is a hint that it could be convenient for an implementation to represent G_i by a special symbol, e. g., a null-pointer.

diagonal row We call a row A_i of matrix A diagonal, if $A_{ii} \neq 0$ and $A_{ij} = 0 \; \forall j \neq i$. This term is therefore not applicable to the vector that is equal to A_i by itself but only in context of the matrix A.

Rank of a decoder The current rank of the matrix G (or A).

A.2 Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, August 10th, 2023

Holger Döbler