A stochastic uncertainty model to measure and improve the robustness of tactical systems in ever-changing communication scenarios

Master Thesis

Johannes Franz Loevenich

Matriculation Number

2375612

This work was submitted to the

Institute of Computer Science IV

University of Bonn, Germany

Adviser(s):

Dr. Roberto Rigolin F Lopes
Dr. Paulo Henrique Rettore Lopes

Examiners:

Prof. Dr. Peter Martini and Dr. Matthias Wübbeling

Registration date: 08-04-2021
Submission date: 16-09-2021

In collaboration with the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), Bonn, Germany

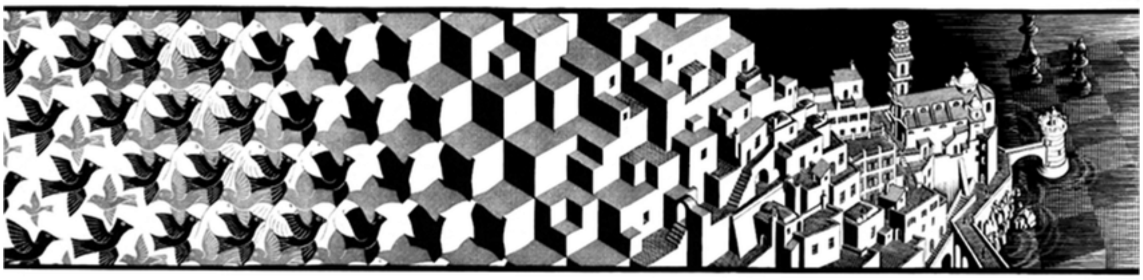


Figure 1 Metamorphosis I by M.C. Escher (May 1937)

Only those who attempt the absurd will achieve the impossible.
I think it's in my basement... let me go upstairs and check.

M.C. ESCHER (1898-1972)

UNIVERSITÄT BONN

Rheinische
Friedrich-Wilhelms-
Universität Bonn

Prüfungsausschuss
Informatik

universität **bonn** · Institut für Informatik · 53012 Bonn

Vorsitzender des Prüfungsausschusses

Prof. Dr. Thomas Kesselheim

Prüfungsamt:
Judith König
Tel.: 0228/73-4418
Fax : 0228/73-4788
pa@informatik.uni-bonn.de
Postanschrift: 53115 Bonn
Endenicher Allee 19a

www.informatik.uni-bonn.de

Erklärung über das selbständige Verfassen einer Abschlussarbeit/ Declaration of Authorship

Titel der Arbeit/Title:

A Stochastic Uncertainty Model to Measure and Improve the Robustness of Tactical Systems in Ever-changing Communication Scenarios.

Hiermit versichere ich, Loevenich, Johannes Franz,
Name / name, Vorname / first name

dass ich die Arbeit – bei einer Gruppenarbeit meinen entsprechend gekennzeichneten Anteil der Arbeit – selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

J. Loevenich
Unterschrift/signature

Bonn, den 15.08.21
date

Abstract

Tactical Communication Systems (TSs) are used to support military forces by facilitating command and control during military operations. Typically, these systems are organized into several layers equipped with cross-layer control mechanisms to handle independent changes from both ever-changing network conditions (*Problem B*) and user data-flows (*Problem A*). Compared to highly qualified military personnel, who have passed years of hardest military training, there is no such methodology for TSs giving quantitative evidence suggesting that these mechanisms can withstand ever-changing communication scenarios. Meaning that the system can handle the user data flow, given the circumstances arising from ever-changing tactical situations and environmental conditions on the battlefield (*Problem* $A|B$). Therefore, this thesis aims to resolve these problems by introducing three stochastic models $Model_B$, $Model_{B^*}$, and $Model_{A|B}$, starting with *Problem B* first and solving *Problem* $A|B$ afterwards. $Model_B$ quantizes the network conditions using a discrete set of states representing the radio modulations supported by tactical radios, thus defining the distribution of changes in the link data rate. The state updates are modeled by a combination of probability distributions defining an in-homogeneous Markov Chain (MC). As a result, the first model establishes a mechanism to generate communication scenarios with varying complexity in a consistent, repeatable and compliant way. In addition, we extend the the Markov $Model_B$ by replacing the transformation functions compositing patterns of data rate changes to communication scenarios by Wasserstein barycentric interpolation to guarantee smooth transitions between patterns by solving the problem of Optimal Transport (OT). As a result, we can transform $Model_B$ to a Markov Decision Process (MDP), where the state space is defined by the cross-distributions resulting from barycentric interpolation between a predefined set of probability distributions. This is a significant improvement, since the reward function of the MDP allows to include system feedback to the modeling process and enables to change the distribution of data rate changes before an experiment is executed. Moreover, we can combine the MDP with an intelligent agent, automating the experiment generation process by learning the distribution for the Time to Resume IP Data Flows (TtR) after unplanned link disconnections, thus defining $Model_{B^*}$. Finally, this thesis introduces a stochastic uncertainty model $Model_{A|B}$ to measure and improve the system's robustness by maximizing the probability of message delivery by proactively adding redundancy. To this end, the model computes the probability of packet/message delivery using cross-layer information within a modern TS with interfaces to the radio, router, message queue, and proxy/gateway. Even though, the model is evaluated within the Tactical Network (TN) domain, it is constructed in a general manner, meaning that it can be applied to arbitrary networks having an IP and message layer, thus improving the reliability of the underlying transport protocols.

Acknowledgments

My personal thanks go to my examiners, namely Prof. Dr. Peter Martini and Dr. Matthias Wübbeling and my advisers Dr. Paulo Henrique Rettore Lopes and Dr. Roberto Rigolin F. Lopes at Fraunhofer FKIE who made this thesis possible in the first place. Their long-term and continuing support, suggestions, motivation, and patience throughout the process of the entire thesis are indispensable.

Moreover, I am very grateful to the student researchers at Fraunhofer FKIE, namely Sharath Maligera Eswarappa and Pooja Hanavadi Balaraju. The cooperation with these extraordinary students considerably affected the quality of this thesis. I also wish to thank my family and friends whose honest opinions, views, and love empowered me at any time during the process of this thesis.

Contents

1

Introduction

The problems addressed in this thesis come from TNs composed of unreliable radio links supporting network-centric Command and Control (C2) services within mobile nodes (e.g. vehicles, dismounted soldiers, unmanned vehicles, and so on). Under the circumstances being due to the military environment to which these networks are deployed, communication scenarios at the edge of TNs are exposed to several sources of randomness [45]. For example, the sources of randomness are node mobility, type of terrain, physical obstacles, weather conditions, radio jamming by an active adversary, physical/cyber attacks, and many more. These random changes considerably affect the robustness of the TS. Nevertheless, TSs need to guarantee reliable communication in scenarios with both user data-flows and network conditions changing independently, also called ever-changing communication scenarios [41, 66]. Given the wide range of military operations, it is challenging to design TSs that can thrive these communication scenarios, also including unplanned link disconnections, which are the worst-case scenario, because the TS should pause the message exchange waiting until the radio link is reestablished or a new route is computed. Therefore, multi-layer control mechanisms have been implemented, within TSs, that rely on feedback from the radio and router computing the current link quality (e.g. data rate, latency, jitter and packet loss) to overcome these issues. The goal of this thesis is to deploy a reproducible mechanism that can be used to test the performance of multi-layer control mechanisms as introduced in [41] and finally improve the robustness of the TS by proactively adding redundancy.

1.1 Research questions and proposed solutions

As mentioned before, TNs form an agile infrastructure on the battlefield, therefore being exposed to extreme variations in the network conditions, which are difficult to anticipate and measure. This fact leads to the first major question of this study:

"How to ensure that simulations/experiments create enough change over radio link data rates to expose the system's limitations while handling unexpected network conditions?"

It should be noted, that this is a less theoretic formulation to what we refer to as *Problem B*, later in Section 3. To answer this question we introduce a stochastic model $Model_B$ to create sequences of network changes (link data rate) defining the distribution of the network conditions of a communication scenario before the experiment is executed. The basic idea of the model is to quantize the network conditions into discrete states, representing the link data rates supported by tactical radios, which are updated by a combination of probability distributions defining an in-homogeneous MC. Moreover, the model implements functions to control the transitions of the MC, which allows "taming the randomness" of the model by trading off the complexity of the underlying probability distribution for partial control of the network conditions of the respective communication scenario. As a result, the model generates different scenarios by generating the changes in the network conditions to test the interoperability among user-facing nodes, tactical routers, and military radios. As part of this thesis, an initial version of the model was published in the IEEE Access Journal [41], therefore the idea and the basic concepts in this thesis remain the same. Nevertheless, the new version of the model is more precise and consistent in the formal definitions, which enables us to use the network conditions generated by the model to measure and improve the system robustness in TNs later on.

Assuming that we have an answer to the first research question, meaning that we can create arbitrary communication scenarios defined by patterns of link data rate changes, we come up with the second major question:

"How to automate the experiment generation process, such that we can include system feedback directly into the modeling process to learn the minimum amount of time that the TS needs to recover from worst-case scenarios (unplanned link disconnections)?"

To answer this question, we enhance the Markov model from [45] to a MDP allowing to introduce system feedback directly into the modeling process. Then the MDP is used to define a multi-agent system learning the distribution of the TtR after unplanned link disconnections (worst-case scenario in military communications), namely $Model_{B^*}$. The resulting distribution describes the minimum time needed for the TSs to recover after these extreme situations.

In a final step, we will discuss one approach to answer the last research question, which can be stated as follows:

"How to measure and improve the robustness of a TS, given that the network conditions and user data flows change independently?"

To do so, we extend the model from [41] to a stochastic uncertainty model describing the system robustness in ever-changing communication scenarios using probabilities for packet and message delivery [39]. Moreover, we state a stochastic optimization problem to optimize the model outputs and compile an algorithm solving this problem by adding a minimum amount of redundancy.

1.2 Hypothesis

This thesis starts with the hypothesis that we can measure the robustness of TSs by developing stochastic models exploiting cross-layer information within a modern TS with interfaces to the radio, router, message/packet queue and proxy/gateway [41, 44, 45, 47], and finally improve the systems robustness by maximizing the probability of message delivery by proactively adding redundancy, as discussed in our previous investigation [39]. Assuming that, the network conditions are quantized by a set of link data rates, we define a multi-agent model to learn the distribution of the TtR user data flows after unplanned link disconnection, thus defining the very first metric to measure the system robustness in worst communication scenarios (unplanned link disconnections). In sequence, we introduce a stochastic uncertainty model describing the robustness of the TS by computing the probability of messaging in the Internet Protocol (IP) and message layer (layer 1 and layer 2) of the TS in arbitrary communication scenarios. Moreover, the model can be used to improve the system's robustness by proactively adding error correction techniques and redundancy (e.g. [80]). Our hypotheses are verified in our test platform [66], using different sets of experiments analyzing the output of the different models introduced in this thesis. Using this approach we show that the set of models can measure and improve the robustness of the system in ever-changing communication scenarios.

1.3 Goals

1.3.1 General

The main subject of this thesis is to develop a reproducible mechanism describing the robustness of TSs in ever-changing communication scenarios. This goal is motivated by the fact that there is no standardized procedure to test TSs as compared to the tough training programs designed for military personnel, before the system is deployed on the battlefield. This thesis aims to fill this gap by defining two models to generate ever-changing network conditions as a combination of patterns of data rates first and introducing a stochastic uncertainty model that is able to measure and improve the robustness of the TS afterwards.

1.3.2 Specific

Since the main goal of this thesis is very extensive, we split the goal into three sub-goals:

- Propose a stochastic model to create ever-changing communication scenarios in TNs:

  As part of this thesis we published two papers, [41] and [45], introducing a stochastic model $Model_B$ to create ever-changing communication scenarios as a sequence of data rate changes generated by an in-homogeneous MC. In this thesis, we revisit this model to upgrade the mathematical theory, thus giving more precise definitions of the basic concepts and more detailed explanations of the functions and algorithms that form the basis of the model.

- Enhance the stochastic model to a multi-agent system composed of an MDP and an intelligent agent allowing to include system feedback directly in the modeling process to learn the distribution of the TtR in worst-case scenarios (unplanned link disconnections):

  To this end, we replace the transformation functions used to “tame the randomness”, therefore allowing for more control of the structure of the patterns of data rate changes, by the concept of Wasserstein barycentric interpolation [6]. The general idea behind this approach is to compile a set of different communication scenarios challenging the TS at different scales ([6, 8]). Since we know the mixed probability distribution of the underlying communication scenarios, we can use Wasserstein barycentric interpolation to define arbitrary smooth cross-distributions slowly converging from not at all (system is robust) to very challenging (system is not at all robust) scenarios. To measure the robustness of the TS, the model is extended to a MDP, which allows to include system feedback from previous experiments directly in the modeling process using a reward function. After the transformation, the resulting $Model_{B^*}$ can be used to measure the robustness of TSs in worst-case scenarios (unplanned link disconnections) by computing the TtR for unicast/broadcast and overlay IP data flows.

- Introduce a stochastic uncertainty model to measure and improve the robustness of TS in ever-changing communication scenarios:

  Finally, we introduce a stochastic uncertainty model $Model_{A|B}$ describing the system robustness using probabilities for packet and message delivery, given arbitrary communication scenarios [39]. Moreover, we state an optimization problem to maximize the probabilities for packet/message delivery and compile an algorithm solving this problem by adding a minimum amount of redundancy packets. As a result, we expect that this concept can help to explore the performance bounds of TSs in ever-changing communication scenarios.

1.4 Thesis structure

The structure of the thesis text is as follows. Chapter 2 discusses the fundamental concepts of military communications and taxonomy in the tactical domain. Moreover, we review recent developments and solutions proposed for similar problems in the literature. Chapter 2 concludes with a motivation coming from both fields: art and science. In sequence, Chapter 3 outlines the main subject of this thesis. Chapter 4 translates the hypothesis into design considerations within an exemplary TS. Also, it discusses the methodology of implementing the stochastic models and solving the stochastic optimization problem to optimize the stochastic uncertainty model, thus describing the robustness of the TS. With quantitative results from a set of experiments generated by the different models, in Chapter 5, the hypothesis is confirmed to show that the models can measure and improve the robustness of the TS in ever-changing communication scenarios. Finally, the observations are summarized and the potential future work is discussed in Chapter 6.

2

Background

This chapter gives an overview of the fundamental concepts used later in this thesis. Starting with a general overview about TNs and a multi-layer architecture for TSs inspired on NATO's Consultation, Command and Control (C3) taxonomy, it follows the definition of ever-changing communication scenarios and a short description of the core services of a tactical middle-ware. Furthermore, the intention of the research question will be motivated by introducing a background of mathematical modeling and a short attempt to connect art and science.

2.1 Military communications

2.1.1 Tactical networks (TNs)

TNs are agile infrastructures consisting of mobile nodes connected via radio links with low data rate and high latency enabling communication to efficiently coordinate and conduct military operations [7, 56, 77]. Thus, the movement of nodes and the presence of physical obstacles on the way will constantly change the network topology and the link quality, which may include link disconnections (worst-case). Moreover, wireless connection and interaction among many mobile nodes can take place spontaneously and in high frequencies, resulting in ever-changing network conditions, such as mobility, resource availability and power constraints. Nevertheless, there is a high demand for robust communication solutions that are easy to deploy wherever needed. Given the high uncertainty, caused by the constraints arising from the challenging environment, this task is hard to complete. This is why it is important to measure and improve the robustness of TSs, if necessary. This study focuses on developing a stochastic model that can describe the robustness of TSs in ever-changing communication scenarios.

There are investigations combining simulation and emulation to create realistic military scenarios [11, 14, 16, 22, 24, 37, 40, 46, 48–52, 60, 62, 71, 76, 77] to test the whole software stack within TSs (also called middleware, proxy, gateway or router) by modeling the network conditions as variations of three fixed states, namely *disconnected*, *intermittent* and *limited*. Even if the state space of the communications scenarios generated by this models is very limited, these investigations lack a precise definition of the distributions of network changes, which make them hard to reproduce. Moreover, most of the experiments represent scenarios in which the changes in the network conditions are generated using non-stochastic means. Therefore, this study is motivated by the fact, that recent literature is missing a well-defined stochastic model to create communication scenarios in TNs, but are also simple to reproduce in a laboratory (e.g. using a channel emulator). Moreover, this model can be extended to a stochastic uncertainty model describing the system robustness by using probabilities.

2.1.2 Tactical communication systems

Tactical communication systems are used to support military units fulfilling the requirements to adapt to ever-changing tactical situations and environmental conditions, thus facilitating command and control of mobile forces. This study follows the NATO's Consultation, Command and Control (C3) taxonomy shown in Figure 2.1 to define three main problems need to be solved to guarantee that the TS enables robust communication under these challenging circumstances. The left side of the figure shows the five functional blocks and the *core services* of NATO's C3 taxonomy [26] forming a hierarchy of queues numbered from (1) to (5). Moreover, the right side of the figure shows the three main problems defining the requirements for a tactical communication system, the ever-changing user behaviour (A), ever-changing network conditions (B) and handling (A) given (B); these definitions are reused/extended from our previous articles reported in [39, 41, 45]. Altogether, the figure shows seven components defined as follows:

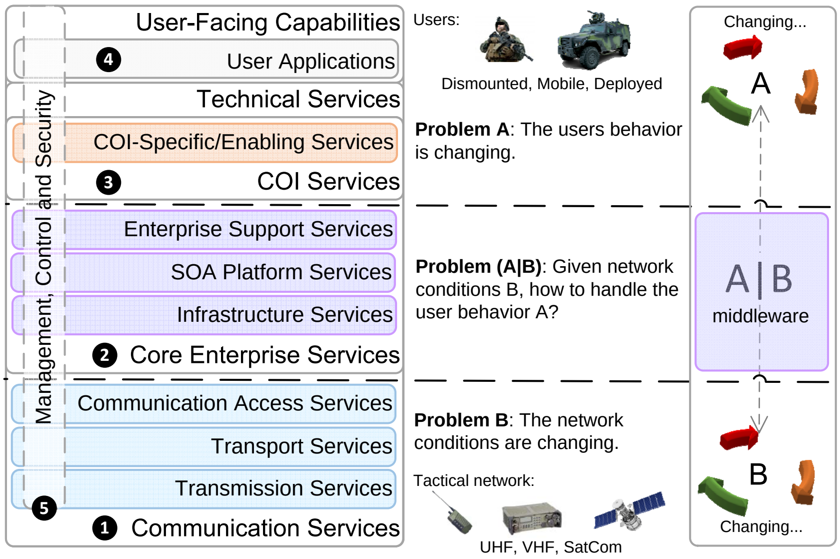

User-Facing Capabilities
4 User Applications
Technical Services
COI-Specific/Enabling Services
3 COI Services
Enterprise Support Services
SOA Platform Services
Infrastructure Services
2 Core Enterprise Services
Communication Access Services
Transport Services
Transmission Services
5
1 Communication Services
Management, Control and Security
Users:
Dismounted, Mobile, Deployed
Problem A: The users behavior is changing.
Problem (A|B): Given network conditions B, how to handle the user behavior A?
Problem B: The network conditions are changing.
Tactical network:
UHF, VHF, SatCom
Changing...
A
A|B
middleware
B
Changing...

Figure 2.1 C3 taxonomy and the three problems [41]

(**A**) *User Application(s)*: define a quantity of QoS-constrained C2 services provided to the users of the TS (e.g. the services in [19]). *Problem A* defines the challenge of creating ever-changing sequences of QoS-constrained messages that can be reproduced for quantitative comparisons. This problem is not addressed in this thesis, but we keep the name convention for consistency with previous work done by the same team at Fraunhofer FKIE;

(5) *Management, Control and Security*: a service sharing cross-layer information among services to compile their current status, to enforce security and to provide feedback to control loops;

(3) *Community of Interest (COI) Services*: hosts services to deal with particularities from the user applications (A) in a collaborative environment, such as coalition among different nations;

(2) *Core Enterprise Services*: a combination of infrastructural services, Service Oriented Architecture (SOA) and enterprise support implemented within TSs.

(1) *Communication Services*: define a combination of services for transmission, transport and network access. Some of them may be implemented by software-defined radios connected with tactical routers managing the TN;

(0) *Radio(s)*: the set of interfaces to the military radios defining the TN. Usually implemented with military protocols like Simple Network Management Protocol (SNMP), Dynamic Link Exchange Protocol (DLEP) [65] and OpenFlow to control and share network metrics in run time;

(**B**) *TN*: an agile infrastructure consisting of mobile nodes connected via radio links with low data rate and high latency enabling communication to efficiently coordinate and conduct military operations deployed in hostile environments, as defined earlier in Section 2.1.1. *Problem B* defines the challenge of creating communication scenarios consisting of ever-changing sequences of link data rates representing the changes in tactical situations and varying environmental conditions (details in Section 4.1).

In the following, this study goes step by step solving *Problem B* first by introducing a stochastic model to create ever-changing sequences of data rate changes representing the varying network conditions in TSs. Thereafter, the model will be adapted to solve *Problem A* given *Problem B* by computing the probability of message delivery given network conditions (B) and adding a minimum amount of redundancy to improve the robustness of the TS.

2.1.3 Ever-changing communication scenarios

Following the definition of Lopes et al. [42], an ever-changing communication scenario in a TN is defined as the combination of user's behavior (*Problem A*) and independently varying networks conditions in a mobile node over time (*Problem A*) (Figure 2.1). As in most of the cases, the main interest of this study is to solve *Problem A* given B, thus showing that the TS can handle the user data flow given the network conditions. In order to solve this problem researchers have developed several models to create challenging communication scenarios that can be seen as test or training environments for the hardware components of the TS, that are comparable to combat fitness tests for human forces.

In [21], the authors propose a solution to estimate or predict the current network state given a set of system parameters defining the performance of the system. Even if, the solution works well for deterministic network conditions, there is no evidence that it is also working over non-deterministic network conditions including link disconnections. The authors in [69] introduced a messaging application that is evaluated in a tool creating network simulations and network field emulations in TNs. Their solution uses a proactive routing protocol to exploit the information from the current system state used as an input for a forwarding algorithm at the application layer called *GetCloser*. Moreover, the define a set of metrics characterizing and quantifying the system performance in several simulations and real experiments.

There are also different studies [53, 54] using real network conditions captured from a military operation of a convoy with few vehicles moving through a particular area in order to test or propose services and applications. The drawback of these studies is, that real deployments are very cost-intensive and difficult to administer, because it demands trained personnel and the military equipment [57]. Therefore, these experiments are difficult to reproduce, and are limited to a specific evaluation scenario. Moreover, there is no precise definition of the scenario, which makes it hard to quantify the performance bounds of TSs using these network setup. Johnson et al. [31], defined hybrid experiments as a combination of real and simulated/emulated network environments to evaluate the system performance by varying the network conditions using non-stochastic means to increase the accuracy quantifying performance metrics and determining when the system fails to perform.

In general, most studies show limited quantitative evidence evaluating TSs over non-deterministic network conditions. Thus, Lopes and Rettore et al. developed models to create ever-changing communication scenarios including the element of chance that can be reproduced for quantitative comparisons. In [42], [41] and [45], the authors introduce an in-homogeneous Markov model and discuss the results over different communication scenarios generated by connecting different patterns of data rate changes, hence proposing to solve *Problem B* [3, 41, 43]. The main idea of the enhanced version of the model is to tame randomness by defining three different functions controlling the state changes of the outer MC of the model and thus controlling the distribution of data rate changes over time. This enables the user to interfere with the model behaviour incorporating the knowledge from previous experiments to generate a set of challenging test scenarios that are simple to reproduce in a laboratory. Another study [66] address the challenge proposing a Tactical Network Test (TNT) platform that creates communication scenarios generated by a stochastic model or mobility models using a reproducible test methodology. The seven scenarios are evaluated in a VHF network by sending uniformly distributed data flows consisting of messages build by IP packets.

Even though these models can be seen as a very first starting point to evaluate the system performance of TNs, they lack precise mathematical and physical foundations explaining the model behaviour as function of time. This study is inspired by these models and aims to improve these solutions for solving *Problem B* by adding more levels of abstraction, meaning that it introduces mathematical concepts explaining the system behaviour by physical equations. This solution for *Problem B* will be evaluated over different user data flows aiming to solve *Problem A* given *Problem B*. This can be done by deploying a middle-ware consisting of in/out chain and control plane to the service taxonomy by placing in/out and control points within the three main services of the NATO's Consultation, Command and Control (C3) Service taxonomy [27] as illustrated in Figure 2.1. The core services of the tactical middle-ware are as follows: The *message queue* is used to decide the admission time between the layers based on contextual information. Then the *UDP transport* fragments the message into UDP packets, which are transferred by the *packet handler* to the radio buffer if a given threshold is not reached. Using *contextual cross-layer monitoring* we gather the network topology and radio buffer occupancy details shared by the *radio plug-in*. Neighbor discovery is performed by Optimized Link State Routing (OLSR) protocol for neighbor discovery and compiles its findings into a routing table.

2.2 Related works

2.2.1 Transport protocols in tactical networks (TNs)

Reliable communication is one of the fundamental requirements for any military network. Existing transport protocols are designed upon a set of assumptions that are violated by the constraints of wireless TNs. To this end, there have been many attempts to adapt reliable protocols like TCP to handle the challenging network conditions in radio links at the edge of TNs [10]. Other protocols try to exploit the topology of military communication scenarios, where most of the nodes move in small groups, developing group communication protocols [75].

Most studies focus on proactive and adaptive configurations of multi-layer architectures deployed in TSs [36, 82]. The idea is to adaptively reconfigure operating parameters in the corresponding layers through proactive prediction based on simple statistics. The information is shared and enriched cross-layer wise to improve the reliability of the TS. In addition, error recovery schemes help to cope with potential route failures caused by link disconnections.

In [58] the authors propose a mathematical model for cross-layer design, which optimizes the trade-offs among different configurations of Software Defined Radios (SDRs) to achieve a maximum performance in terms of energy efficiency, reliable packet delivery and within affordable latency bounds in multi-hop TNs.

These investigations motivated us to develop a stochastic uncertainty model that is based on reproducible communication scenarios generated by the stochastic model described in [45]. As a result, the outcomes and metrics of this model are more explicit and we can control the behaviour and the learning process of our model by slightly increasing the complexity of the underlying communication scenario.

2.2.2 Robust routing using reinforcement learning

Reinforcement Learning (RL) has become a major technique to increase the robustness of routing protocols in dynamic Mobile Adhoc Network (MANET)s. Early works proposed Q-learning using a RL module to route packets with very low delivery time ([9],[78]). This approach exploits node statistics for each destination and next-hop pairs in the network, which are also called Q-values. These values represent the quality of each possible next-hop or network route and this is why they are stored and updated in a Q-table. The resulting Q-routing protocol sends the data packets to the next hop (unicast) with the best Q-value.

Since Q-learning is efficient in low dynamic and static networks only, the framework was improved for more dynamic networks in [35]. The basic idea is to add a confidence level table (C-values) which are incremented each time the Q-value is updated and decremented when gets outdated. The drawback of the so called CQ-routing technique is that it becomes inefficient in highly dynamic networks. This is because it is based on unicast for single nodes, which makes the updates of the Q-values time consuming in communication scenarios with rapid network changes.

In [32] the authors introduce the Smart Robust Routing (SRR) algorithm, which combines RL with an adaptive routing (AR) scheme [15]. This allows to dynamically switch between unicast and broadcast to improve robustness in TNs, while reducing the total network overhead. To this end, the authors use a combination of Q-factors to collaboratively learn the shortest-path to the destination and C-factors representing a likelihood of reaching the destination given a respective next hop plus a broadcast procedure ($CQ+$ routing) for high reliability, robustness and network exploration. The resulting SRR protocol has been evaluated in a random mobility scenario with 30 nodes randomly deployed in an 800m by 200m area using the CORE framework [1]. In this scenario, the source and destination are static while, the other nodes in between move randomly with varying speed. The source is transmitting data at a rate of ten 1000-byte packets per second (80kbps). The performance has been evaluated using goodput (packet receiving rate at the destination) and total overhead (network load). Using these metrics the authors showed that SRR manages to provide very little packet loss ($< 2\%$) at only a slight increase of overhead compared to other methods.

In [34] the authors enhanced the $CQ+$ routing applying a multi-agent deep reinforcement learning (MADRL) approach to design a robust, reliable, and scalable policy for MANETs. The focus of the deep reinforcement learning (DRL) framework is on scalability, meaning that the routing policies can be trained and tested using variable network size, data rate and mobility dynamics. It is shown that this routing protocol is much more efficient than $CQ+$ routing when comparing the normalized total network overhead. In addition, the resulting shared policy of the model is scalable, which makes it possible to reuse the policy in different scenarios.

The theory and recent investigations in RL, especially related to MANETs and TNs, motivated us to enhance $Model_B$ from [45] to a MDP whose optimal policy is computed using RL. Since the long-term goal is to deploy an intelligent model, which is capable of learning the performance bounds of TSs without human supervision, we decided for a more formal definition of the RL task by defining a Markov decision process. This allows for the possibility of introducing system parameters directly into the underlying Bellman equations of the MDP, hopefully enabling the system to learn arbitrary performance metrics of the TS in a future investigation.

2.3 Motivation: Art and science

The models introduced in this thesis were inspired by physics (Quantum mechanics) and art (M. C. Escher drawings [38]), which can both be described using mathematics. Humans have created mathematical and physical principles to model and explain the universe that we all are living in. These principles equip us with the tools to explain natural phenomena extending from the random behaviour of quarks to the movement of planets on orbits. But, the most fascinating thing is that the language of mathematics is that it is unreasonably effective. Or more precise, as Eugene Wigner said: *"The miracle of the appropriateness of the language of for the formulation of the laws of physics is a wonderful gift which we neither understand nor deserve"* [81]. Even if there is the possibility that we never understand the origin of these wonderful gift, we can get it to reveal its hidden secrets by following its rules. For sure, creativity, skill and courage are required, but it is worth it.

But where does the whole process of creation start? It all starts in our mind. Each argument, logical implication or transformation of an equation starts as a thought in our brain. And that is why there has always been a close connection between art and science. Both, artists and scientists are driven to observe and create. For example, Leonardo da Vinci was skilled across the arts and sciences. As Daniel Smith says in his book *"How to Think Like da Vinci"* [23]:

"We wonder how one man could be so skilled across the arts and sciences. The answer is that he recognized no intellectual separation between his work as an artist and as a scientist. Instead the art and the scientist were conjoined, their ideas flowing effortlessly together informing his practice in whatever discipline he happened to be focusing upon on any given day... the Mona Lisa could not have been painted had he not devoted countless hours to the study of anatomy."

Moreover, there is no natural boundary for the process of creation although both our brain and our computers are bounded by the fundamental laws of physics. Sometimes it may feel like we are missing the tools for solving a specific problem, but we should always keep in mind that we are capable to create the tools by using the language of mathematics. No matter, how hard the problem if we want to create wonder, we are set for action. Following F. Nietzsche, this is why we need art:

"Art sets the bounds for wonder, so humans are set for action. Once in motion, humans are no longer artists but a work of art."

Like many other scientific works in the past 60 years, this study is impressed with the graphic art of M. C. Escher [38], recognizing with fascination that a great number of Escher's images relate directly to many scientific and mathematical principles. More particularly, we are interested in two series of pictures called metamorphosis (see Figure 2.2) and circle limit (see Figure 2.4). The concept of the metamorphosis is to morph one image into a tessellated pattern, then gradually to alter the outlines of that pattern to become an altogether different image. Figure 2.2 shows the metamorphosis II of the three-part series. In this work Escher begins with the word "metamorphose" which is then transformed into different patterns of rectangles until they are forming a grid. This grid then becomes a black and white checkered pattern, which is transformed into a tessellations of animals like reptiles, a honeycomb, insects, fish, birds until it is modified into a pattern of blocks with red tops.

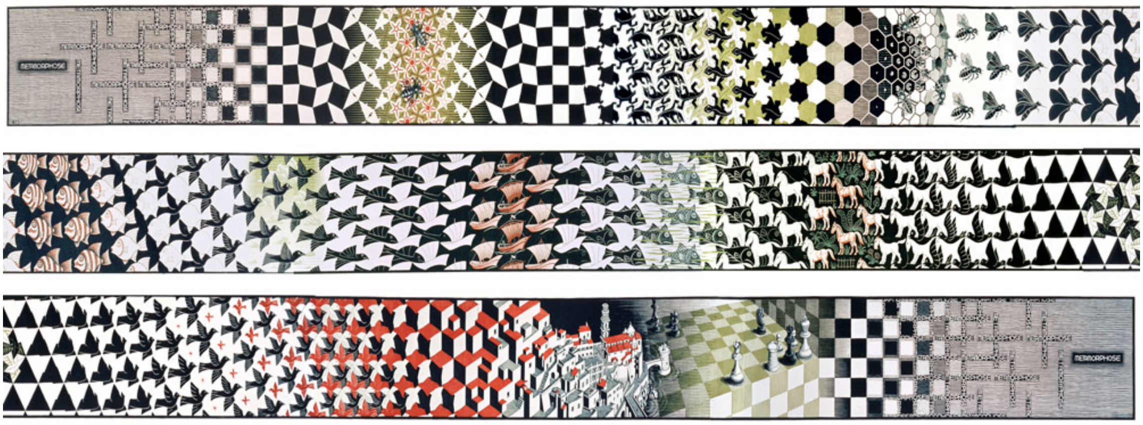
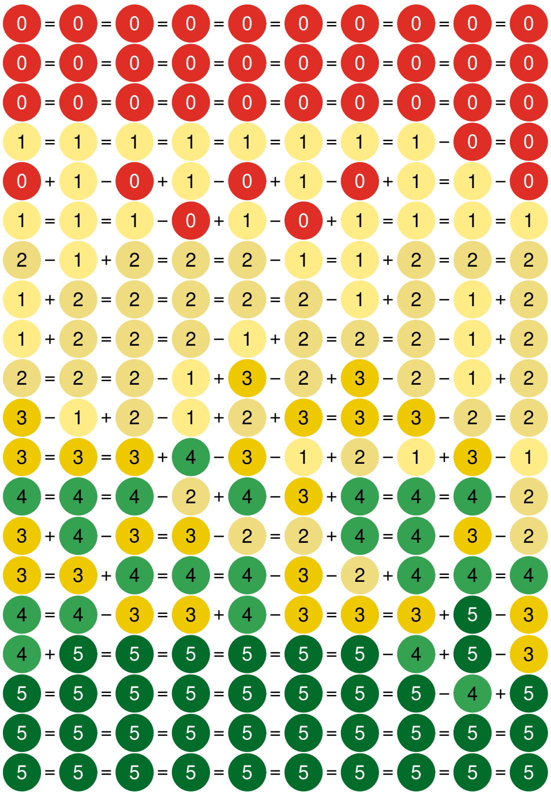


Figure 2.2 Metamorphosis II by M.C. Escher

Finally the blocks become the image of the Italian town Atrani, which is linked by a bridge to a chess rook piece. The water between the town of Atrani and the rook then becomes a chess board with other chess pieces. In the last part of the series the chess board is modified into a grid of black and white, which becomes the word metamorphose again. This concept inspired us to create communication scenarios in TNs by using a stochastic model that creates communication scenarios as succession of different patterns of data rate changes. As a result, a communication scenario can be seen as a mixed probability distribution created by an in-homogeneous MC.



```
0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0
0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0
0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0
1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 − 0 = 0
0 + 1 − 0 + 1 − 0 + 1 − 0 + 1 = 1 − 0
1 = 1 = 1 − 0 + 1 − 0 + 1 = 1 = 1 = 1
2 − 1 + 2 = 2 = 2 − 1 = 1 + 2 = 2 = 2
1 + 2 = 2 = 2 = 2 = 2 − 1 + 2 − 1 + 2
1 + 2 = 2 = 2 − 1 + 2 = 2 = 2 − 1 + 2
2 = 2 = 2 − 1 + 3 − 2 + 3 − 2 − 1 + 2
3 − 1 + 2 − 1 + 2 + 3 = 3 = 3 − 2 = 2
3 = 3 = 3 + 4 − 3 − 1 + 2 − 1 + 3 − 1
4 = 4 = 4 − 2 + 4 − 3 + 4 = 4 = 4 − 2
3 + 4 − 3 = 3 − 2 = 2 + 4 = 4 − 3 − 2
3 = 3 + 4 = 4 = 4 − 3 − 2 + 4 = 4 = 4
4 = 4 − 3 = 3 + 4 − 3 = 3 = 3 + 5 − 3
4 + 5 = 5 = 5 = 5 = 5 = 5 − 4 + 5 − 3
5 = 5 = 5 = 5 = 5 = 5 = 5 = 5 − 4 + 5
5 = 5 = 5 = 5 = 5 = 5 = 5 = 5 = 5 = 5
5 = 5 = 5 = 5 = 5 = 5 = 5 = 5 = 5 = 5
```

Figure 2.3 Pattern of link data rate change from [45]

Figure 2.3 from [45] illustrates a communication scenario as sequence of patterns starting with the best network conditions in $\bar{B}_{5\text{(green)}}$ and goes towards the worst conditions in $\bar{B}_{0(\text{red})}$ by jumping through B_4, B_3, B_2 and B_1. Interpreting the best conditions in $\bar{B}_{5(\text{green})}$ as the word metamorphosis and the worst conditions in $\bar{B}_{0(\text{red})}$ as the town of Atari we can link this to Figure 2.2 by jumping through B_4, B_3, B_2 and B_1 first and then jumping all the way around through B_1, B_2, B_3 and B_4 in sequence; notice that the changes occur from bottom to top (vertically) instead of horizontally. In this case we fix the order of the patterns to tame randomness by playing M.C. Escher with patterns of data rate changes. This process can be extended to create more extreme scenarios by defining a probability distribution for the pattern transitions. Figure 4.5b shows a communication scenario using the same patterns as Figure 4.5a, but each line is generated by sampling a pattern from $\bar{B}_0, B_1, B_2, B_3, B4$ and $\bar{B}_0$ uniformly at random. The advantage of using a probability distribution comes with the fact of trading control of the model for complexity of the scenarios. Meaning that, the random behaviour of the model makes it hard to define a series of communication scenarios using the history of experimental results to decide the next one.

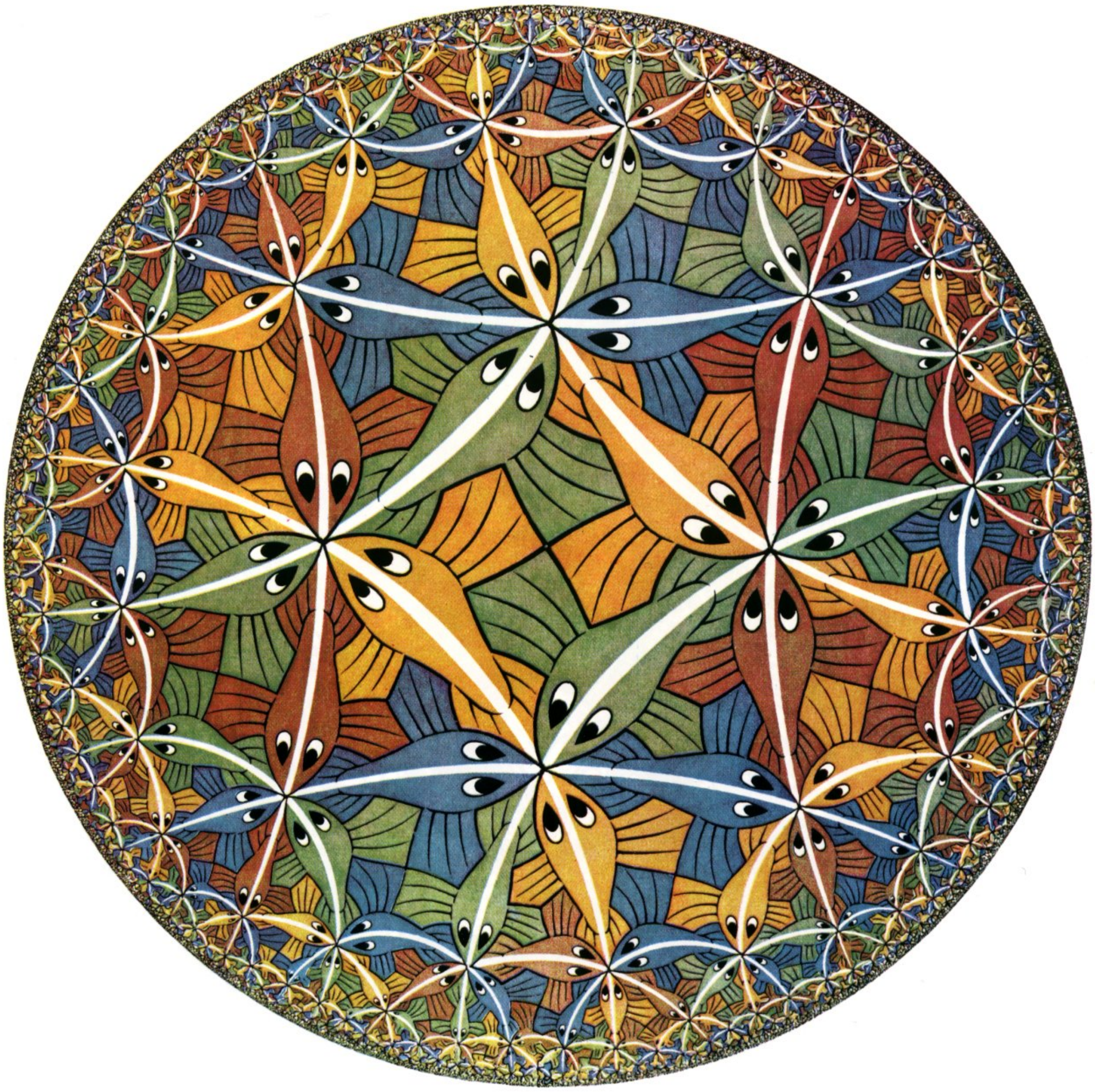


Figure 2.4 Circle limit III by M.C. Escher

This raises the following question: Given the distribution of two different communication scenarios $\mathcal{C}_1$ and $\mathcal{C}_2$ and the number of tessellated patterns β, how to find the cross-distributions of the patterns defining the metamorphosis from $\mathcal{C}_1$ into $\mathcal{C}_2$? Now one could argue that there are arbitrary many ways to define the set of cross-distributions, which is true if no preconditions must be fulfilled. Taking M.C. Escher as inspiration again, one can argue that the beauty of the metamorphosis comes from the fact that the geometrical shapes in Figure 2.2 easily merge into each other. This can be achieved for $\mathcal{C}_1$ and $\mathcal{C}_2$ by raising the condition that two elements of the series of cross distributions always needs to be nearest neighbours with respect to the Wasserstein metric defined in [6]. Given a start scenario $\mathcal{C}_1$ this approach allows to generate arbitrary smooth pattern transitions morphing $\mathcal{C}_1$ into $\mathcal{C}_2$ by using Wasserstein barycentric interpolation [6] and solving the mathematical problem of OT [12, 13, 20, 70].

The key idea behind this approach is that one can break the transformation of two communication scenarios $\mathcal{C}_1$ to $\mathcal{C}_2$ into a set of patterns such that the difference of two consecutive patterns goes to zero as β goes to infinity. The hope is that minor changes in the patterns also result in minor changes of the features that are monitored from the different layers of the TS shown in Figure 3.1. This is close to the so called infinity principle by Strogatz [74]. The principle says that to analyze something complicated, you should first break it down into an infinity of simpler parts and analyze those. Again, there is also a work of M.C. Escher illustrating the process of infinity and thus connecting the ideas of art and science that define the main inspiration of this study. Figure 2.4 illustrates the circle limit III woodcut of M.C. Escher, where “strings of fish shoot up like rockets from infinitely far away” and then “fall back again whence they came” (Coxeter 1979).

Once we are able to create communication scenarios that are capable of representing the variation of military communication on the battlefield, we are interested in learning the performance bounds of TSs over these scenarios. To this end, the stochastic models measuring and finally improving the robustness of TS are inspired by intelligent systems that can learn without human supervision [73]. Even if the time frame of this thesis does not allow to develop such a complex model, this study can be seen as a starting point introducing the basic concepts defining the baseline for the intelligent system. So let us move on with the words of M.C. Escher and “attempt the absurd to achieve the impossible”, thus creating stochastic models to measure and improve the robustness of TSs, keeping always in mind that the overall goal is to define a model that can learn the performance bounds of TSs without human supervision.

3

Problem statement

This chapter compiles the problem statement motivated by the challenges of modern TNs, art, and science as explained in the last chapter. For simplicity, we divide the problem into three sub-problems A, B and $A|B$, inspired by the layer-wise architecture of TSs. Here, *Problem A* is related to the ever-changing user behaviour during military communication scenarios, *Problem B* to the ever-changing network conditions resulting from the circumstances being due to the environment TSs are deployed, and *Problem $A|B$* is a combination of both problems. To define the problems formally, we start with explaining a basic military end-to-end communication scenario first and go step by step defining all three problems afterwards.

3.1 End-to-end communication scenario

Remembering that modern TSs are organized into several layers and equipped with multi-layer control mechanisms to handle independent changes from both user data-flows (*Problem A*) and network conditions (*Problem B*), Figure 3.1 shows an exemplary end-to-end communication scenario with the *sender* and the *receiver* connected through a radio link, composing of an ever-changing communication scenario [39, 41, 45]. In this setup, each node has a *control plane* (c) and two chains: one for *incoming* (i) data-flows and another for *outgoing* (o) data-flows, both sitting in at least four layers, namely radio (*layer 0*), packet (*layer 1*), message (*layer 2*) and proxy/broker (*layer 3*). Notice the connection symmetry through a noisy radio channel, where the *out* chain from the sender is connected to the *in* chain of the receiver and vice-versa.

The sequence of messages from command and control systems (A) enter the system from *layer 3* carrying a set of Quality of Service (QoS) requirements such as priority, reliability and time of expire (differentiated at *layer 2*), which are partially mapped to IP packets at *layer 1*. The radio (*layer 0*) usually has a buffer with limited size that differentiates the IP packets by priority. Note that a multi-homed node

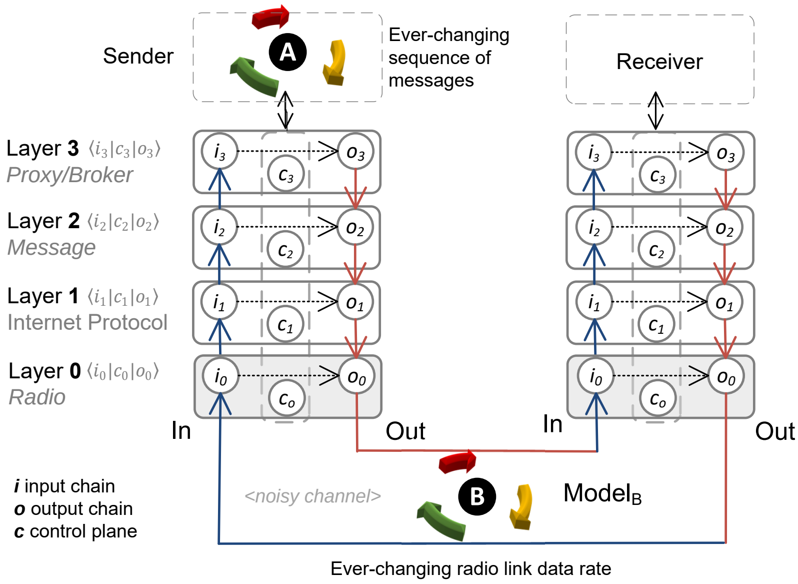


Figure 3.1 Ever-changing end-to-end communication scenario [39]

with r radio networks will have r instances of this hierarchy of queues to handle the difference in both coverage (kilometers) and link data rate (kbps) from military communication technologies, such as High Frequency (HF), Very High Frequency (VHF), Ultra High Frequency (UHF) and Satellite Communications (SatCom).

3.2 Problem definition

With the explanation of a military end-to-end communication scenario in mind, we can move further and define the problems addressed in this thesis formally. As mentioned before, we focus on solving problems B and $A|B$. Thus, we assume the user data flow (*Problem* A) to be uniformly distributed. For the sake of completeness, however, we state *Problem* A formally as follows.

Definition 3.2.1 (*Problem* A)**.** *Given a set of C2 systems available to military users, define Problem A to be the problem to create ever-changing user data flows representing the variety of military communication in the network.*

3.2.1 *Problem* B: Modeling ever-changing link data rates

One key prerequisite to develop and validate a stochastic model improving the robustness of TSs (i.e. solution to *Problem* $A|B$) is the ability to create experiments representing the variety of possible communication scenarios in TNs (i.e. solution to *Problem* B).

Definition 3.2.2 (*Problem* B)**.** *Given a radio device supporting a set of modulations, define Problem B to be the problem to create ever-changing network conditions by quantizing the network states using a discrete and finite set of data rates, supported by the radios of the TS.*

To this end, the first goal is to define a stochastic model $Model_B$ to create sequences of link data rates representing the network conditions of TS over a finite time horizon $\mathfrak{T}$, which can be used to compute its robustness to changes in the network in a second phase (solving *Problem* $A|B$).

Definition 3.2.3 ($Model_B$)**.** *Given a radio device supporting a set of modulations, then* $Model_B$ *is defined to be a stochastic model capable of solving Problem B by creating ever-changing sequences of link data rates, including link disconnections of arbitrary duration.*

In this thesis, we define robustness to be the system ability to cope with both run-time errors and erroneous input [30]. These errors can be caused by a variety of reasons. One reason might be that the system is not capable to identify the changes in the link metrics, such as data rate, latency, jitter and packet loss. This could be the case if the system is lacking an interface to the radios and tactical routers, or multi-layer control loops to cope with changes in the network. This is why our team at Fraunhofer FKIE developed multi-layer control loops, that rely on cross-layer contextual monitoring, thus analyzing the network metrics and giving quantitative feedback to multi-layer control mechanisms. The control loops actuate in enforcement points adapting both incoming and outgoing data-flows [3, 41–43]. Nevertheless, it remains the problem to give quantitative evidence that these control mechanisms can handle ever-changing network conditions as they occur in TN.

Moreover, there is, for the time being, no option to include system feedback directly in the modeling process of the network conditions, which enables for automated execution of experiments by exploiting the metrics learned from previous experiments. This is, why we propose an enhanced version $Model_{B^*}$ of $Model_B$ from [45] transforming the in-homogeneous Markov model to a MDP (defined later in Chapter 4). As a result, the MDP allows to include system feedback by a real valued reward function. Then, we can combine the MDP model with an intelligent agent to learn a target metric, which enables to automate the modeling process. In this thesis, the target metric is defined to be the TtR after unplanned link disconnection representing the worst event in any communication scenario.

3.2.2 *Problem* A|B: Modeling robust tactical systems (TSs)

Given a model solving *Problem B*, thus generating the network conditions $\mathcal{N}$ of a communication scenario $\mathcal{C} = (\mathcal{N}, \mathcal{U})$, the goal is to develop a stochastic uncertainty model that is able to describe and finally improve the robustness of the TS using probabilities for packet and message delivery [39]. In other words, we are interested in a model that can give quantitative evidence if the TS can handle the user data flow $\mathcal{U}$, given the network conditions $\mathcal{N}$.

Definition 3.2.4 (*Problem* $A|B$)**.** *Given a TS organized into several layers and equipped with multi-layer control mechanisms, define Problem* $A|B$ *to be the problem to handle ever-changing user data flows* $\mathcal{U}$ *(Problem A), while the radio link data rates defined by* $\mathcal{N}$ *are following a set of unknown probability distributions (Problem B).*

Assuming that the user data flow $\mathcal{U}$ is composed of uniformly distributed messages, we compute the probability of a message being successfully transmitted from *sender* to *receiver* in Figure 3.1, while the radio link is changing according to the distribution defined by $\mathcal{N}$. As comparing to the layer-wise architecture of the TS, we can compute the probability of message delivery (layer 2) by computing the probability of packet delivery in the IP layer (layer 1). This is done by implementing a technique called *Reed Solomon Code* ([25, 80]) and proactively adding redundancy, as explained later in Section 4.3. In short, the final goal of this thesis is to define a $Model_{A|B}$ that is defined as follows.

Definition 3.2.5 ($Model_{A|B}$)**.** *Given a TS organized into several layers and equipped with multi-layer control mechanisms, then* $\text{Model}_{A|B}$ *is a stochastic uncertainty model capable of solving Problem $A|B$. In other words,* $\text{Model}_{A|B}$ *can measure and improve the robustness of the TS for Problem $A|B$.*

3.3 Objective

The goal of this study is to solve both problems B and $A|B$ by defining three stochastic models, two generating the network conditions $\mathcal{N}$ and thus solving *Problem B* and another one solving the uncertainty *Problem $A|B$* finding a close to optimum configuration, thus improving the system robustness by adding a minimum amount of redundancy. We start by enhancing the model from [45] to create sequences of link data rates to quantify the performance bounds of TSs. Based on that, we define the stochastic uncertainty model describing the robustness of the TS by computing the probabilities for packet and message delivery for arbitrary communication scenarios generated by the model solving *Problem B*. We conclude this study by investigating optimal solutions (Chapter 4) for the uncertainty model over a set of experiments generated in a VHF network (Chapter 5).

4

Design

In this chapter we define the models to solve both *Problem B* and *Problem $A|B$*, earlier described in Chapter 3. The three models, namely $Model_B$, $Model_{B^*}$ and $Model_{A|B}$ are stochastic, the first one generating the network conditions of communication scenarios at the tactical edge, the second one improving $Model_B$ to a multi-agent model allowing to include system feedback directly in the modeling process and the third model describing and finally improving the system robustness by adding redundancy at the IP layer.

This study started with the hypothesis that we can quantify the network conditions of a TN into discrete states representing the link data rates supported by the tactical radios in a laboratory environment ([45]), because we had access to laboratory with real military radios at Fraunhofer FKIE. The motivation behind this approach comes from quantum physics and the idea that a lot of effects and phenomena in this universe can be described by quantities that can take only discrete values. Thus, $Model_B$ represents an attempt to quantify the network conditions of TN in the sense of information theory. Coming from this smallest quantities, these can be combined to create patterns of data rate rate changes updated by a combination of probability distributions over a finite time horizon. Moreover, we can increase the level of complexity by combining these patterns of data rate changes as a function of time or by probability distributions again, called ever-changing network conditions. This enables us to generate complex sequences of data rate changes to analyze the robustness of TS over ever-changing network conditions.

Later on, it became clear that this approach was missing the opportunity to include system feedback directly in the modeling process, thus allowing for an automated experiment execution. This was an significant disadvantage, since the analysis and reconfiguration of the transformation functions had to be done manually. To solve this problem, $Model_{B^*}$ transforms $Model_B$ to a MDP, which allows to include the system feedback in the reward function of an agent. Then, we define a second, intelligent agent, thus extending the model to a multi-agent system to learn a predefined target metric. In other words, the metric defines the target quantity for the analysis of the features captured from the monitoring service during an experiment and is included into the experiment generation process using the reward function of the MDP. Remembering, that the optimal policy of an MDP is a MC and thus $Model_B$ can be seen as an instance of $Model_{B*}$. As a result, we can automate the process for solving *Problem B* by defining the metric to be learned by a multi-agent system composed of a MDP and an intelligent agent.

Finally, $Model_{A|B}$ exploits the layer-wise architecture of TS to measure the robustness of the system by computing the probability of packet/message delivery in the IP and message layer. To this end, $Model_{A|B}$ computes the probability of packet and message delivery in the IP and message layer of the TS, thus defining an uncertainty model to describe the robustness of the TS, while the network conditions of the system behave according to the sequences of data rates generated by $Model_B$ or $Model_{B^*}$ respectively. Note that all algorithms and methods described in this section were implemented in *Python* and also deployed to a TS hosted in a laboratory at Fraunhofer FKIE. The experimental results of the models are presented and discussed comprehensively in Section 5.

4.1 Model B: Creating ever-changing link data rates

In this section, we develop the stochastic $Model_B$ to create the network conditions of ever-changing communication scenarios represented by sequences of link data rates, thus solving *Problem B*. The model is composed of a set of patterns of data rate changes $\mathcal{P}$ and two distributions over a finite time horizon $\mathfrak{T}$, one representing the link data rate changes in a single link and another one defining the length of the time interval for updating the data rate . More precisely, the distribution of link data rate is defined by the MC B_{outer} and the time distribution for state update by λ, in short $Model_B(\mathfrak{T},\mathcal{P},B_{\text{outer}}, \lambda)$. The original motivation for the introduction of a stochastic model to create ever-changing sequences of link data rates is that, by now, there is no way to generate test scenarios quantifying the robustness of military communication in a controlled environment (i.e. a laboratory at Fraunhofer FKIE) at arbitrary scale. This section gives the theoretical foundations to instantiate a model to create reproducible communication scenarios for quantitative comparisons.

Definition 4.1.1 (Instance of $Model_B$)**.** *Given a finite time horizon $\mathfrak{T} = 1, \dots, T$, a set of patterns $\mathcal{P}$ representing distributions of data rate changes, a transition matrix B_{outer} and the time distribution for state updates in the link λ, then* Model_B *($\mathfrak{T}$,$\mathcal{P}$,B_{outer}, λ) is defined to be the in-homogeneous Markov model [29] mapping these inputs to a finite sequence of data rate changes Σ changing as function of time distributed by λ. (Figure 4.1).*

The following sections describe each single component of the $Model_B$ and also explain how to instantiate the model. As referring to Figure 3.1 the model is taking actual influence of layer 0 (Radio layer) of the TS. Doing this, the model changes the link quality between sender and receiver radio according to the sequence of data rates Σ and the link update time distributed by λ.

4.1.1 Instantiating the model

Let us start defining a communication scenario $\mathcal{C}$ at the tactical edge as the combination of of user's behaviour $\mathcal{U}$ and independently varying network conditions $\mathcal{N}$.

Definition 4.1.2 (Communication scenario at the tactical edge)**.** *A communication scenario at the tactical edge $\mathcal{C} = (\mathcal{N}, \mathcal{U})$ is defined as the combination of user's behaviour $\mathcal{U}$ and independently varying network conditions $\mathcal{N}$.*

Assuming that the network conditions $\mathcal{N}$ of the communication scenario can be represented as the composition of patterns of data rate changes, we define a pattern of data rate changes as the 4-tuple $P = (S, \Lambda_P, B_P, \vec{X}_0)$, where S is the state space representing the different data rates supported by the tactical radios, Λ_P is the pattern length, B_P is a stochastic matrix representing the distribution of state transition of a MC as referring to Figure 4.2 and $\vec{X}_0$ is the $|S|$-dimensional initial state vector.

Definition 4.1.3 (Pattern of data rate changes)**.** *Given the pattern length Λ_P, a stochastic matrix B and an initial state $\vec{X}_0$, a pattern of data rate changes P over state space S is defined as the 4-tuple $P = (S, \Lambda_P, B_P, \vec{X}_0)$ over a finite time horizon $\mathfrak{T}_{\Lambda_P} = \{1, \ldots, \Lambda_P\}$. The corresponding sequence of data rate changes $(\sigma_1, \ldots, \sigma_{\Lambda_P})$ according to P can be generated by sampling Λ_P different states from the probability vectors $\mathcal{X} = (X_1, \ldots, X_{\Lambda_P})$, where $X_i \in \mathcal{X}$ is defined as $X_i = X_0^T \cdot B_P^i$.*

For example, Table 4.1 lists two sets of stochastic matrices representing the probability distributions for the state transitions of twelve exemplary patterns with state space $S = \{s_0 = 0.0, s_1 = 0.6, s_2 = 1.2, s_3 = 2.4, s_4 = 4.8, s_5 = 9.6\}$. In this table, $\bar{B}_{0..5}$ (left column) define stable network conditions with no changes in the link data rate and $B_{0..5}$ (right column) combine the six states to incrementally improve the average link data rate, except B_5 defining equal conditional probabilities for all the states ($^1/_6$).

It should be noted, that $\bar{B}_0$ represents the worst-case distribution, meaning that there is a 100% chance that the radio link between both nodes stays disconnected

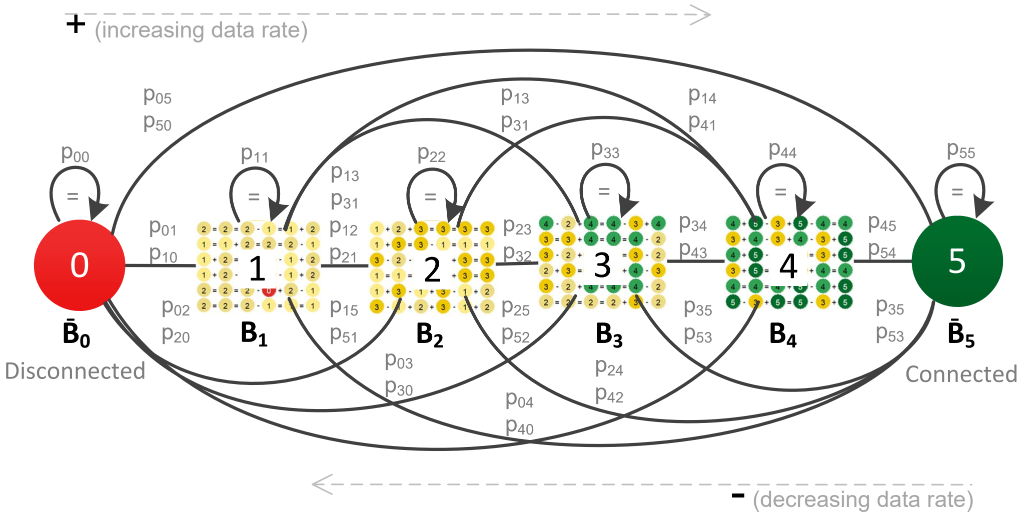


Figure 4.1 Nested Markov chain with patterns as states [45]

$$
\begin{gathered}
\bar{B}_0\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}
\quad
B_0\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1/3 & 2/3 & 0 & 0 & 0 & 0 \\ 1 & 1/3 & 2/3 & 0 & 0 & 0 & 0 \\ 2 & 1/3 & 2/3 & 0 & 0 & 0 & 0 \\ 3 & 1/3 & 2/3 & 0 & 0 & 0 & 0 \\ 4 & 1/3 & 2/3 & 0 & 0 & 0 & 0 \\ 5 & 1/3 & 2/3 & 0 & 0 & 0 & 0 \end{array}
\\
\bar{B}_1\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}
\quad
B_1\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & .01 & .33 & .66 & 0 & 0 & 0 \\ 1 & .01 & .33 & .66 & 0 & 0 & 0 \\ 2 & .01 & .33 & .66 & 0 & 0 & 0 \\ 3 & .01 & .33 & .66 & 0 & 0 & 0 \\ 4 & .01 & .33 & .66 & 0 & 0 & 0 \\ 5 & .01 & .33 & .66 & 0 & 0 & 0 \end{array}
\\
\bar{B}_2\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 5 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}
\quad
B_2\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 2 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 3 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 4 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 5 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \end{array}
\\
\bar{B}_3\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}
\quad
B_3\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & .01 & .33 & .33 & .33 & 0 \\ 2 & 0 & .01 & .33 & .33 & .33 & 0 \\ 3 & 0 & .01 & .33 & .33 & .33 & 0 \\ 4 & 0 & .01 & .33 & .33 & .33 & 0 \\ 5 & 0 & .01 & .33 & .33 & .33 & 0 \end{array}
\\
\bar{B}_4\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}
\quad
B_4\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & .01 & .01 & .32 & .33 & .33 \\ 2 & 0 & .01 & .01 & .32 & .33 & .33 \\ 3 & 0 & .01 & .01 & .32 & .33 & .33 \\ 4 & 0 & .01 & .01 & .32 & .33 & .33 \\ 5 & 0 & .01 & .01 & .32 & .33 & .33 \end{array}
\\
\bar{B}_5\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 & 0 & 0 & 1 \\ 5 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}
\quad
B_5\;\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 2 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 3 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 4 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 5 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{array}
\end{gathered}
\tag{4.1}
$$

Table 4.1 Two sets of stochastic matrices [45]

and $\bar{B}_5$ represents the opposite, keeping the data rate in the link at maximum

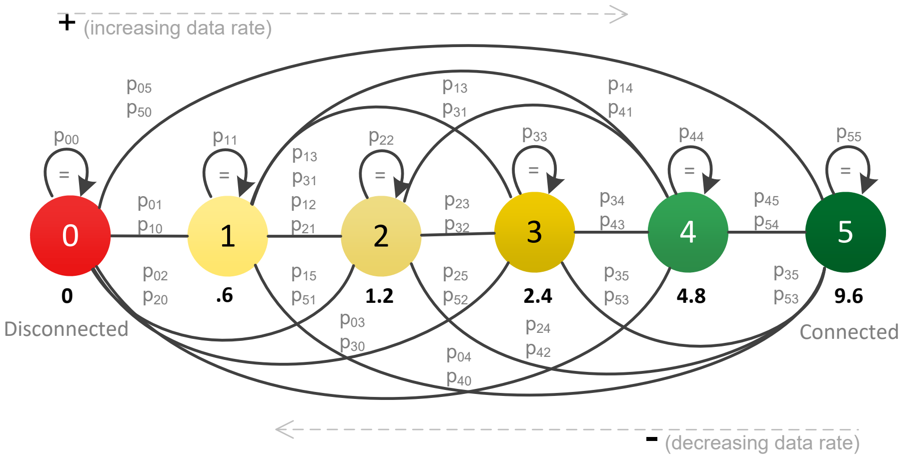


Figure 4.2 Markov chain with link data rates as states [41]

nominal capacity (9.6 kbps). In between, there are ten exemplary patterns creating a range of network conditions from low (< 2 kbps) to high (> 4 kbps) data rates on average.

One basic assumption in this study is that the variation in the link data rate is the most important metric to evaluate multi-layer store-and-forward mechanisms in TS, because latency, jitter and packet loss are by products of fluctuations in the data rate (or vice-versa). Later in Section 5 we discuss quantitative experiments showing how the changes in the link data rate impacts both latency and jitter in a VHF network.

Assuming that the network conditions of every communication scenario consist of a finite number of patterns $(P_t)_{t \in \mathfrak{T}}$ over the time horizon $\mathfrak{T} = (1, \ldots, T)$, define a finite MC $\mathcal{Y}_{\text{inner}} = (X_l)_{l \in \Lambda_P}$ with state space $S_{\text{inner}} = (s_0, \ldots, s_{N-1})$, where the states $s_i \in S$ represent nominal data rates supported by different radio modulations or waveforms. Figure 4.2 shows an exemplary MC for VHF radios supporting six different states $S = (0, \ldots, 5)$: disconnection *state 0*, *state 1* the 0.6 kbps of nominal data rate and so on (for 1.2, 2.4 and 4.8 kbps) until *state 5*, which represents 9.6 kbps.

This construction can be generalized to work with arbitrary radio modulations or waveforms by defining the discrete and finite set S_{inner} as the N different nominal data rates of the radio modulation $S_{\text{inner}} = \{s_0, \ldots, s_{N-1}\}$. For reasons of simplicity, we assume that the set S_{inner} is ordered, meaning that the maximum nominal data rate represents the best scenario s_{N-1} (say state 5 in our case, see Figure 4.2) and link disconnection is the worst scenario s_0 (state 0), because the node is disconnected from the network and can not communicate. With these conditions considered, the Markov model in Figure 4.2 can be used to create sequences of data rates representing an arbitrary pattern $P = (S_{\text{inner}}, \Lambda_P, B_P, \vec{X}_0)$ by sampling Λ_P different states from the MC $\mathcal{Y}_{\text{inner}} = (X_1, \ldots, X_{\Lambda_P})$, where $X_i \in \mathcal{Y}_{\text{inner}}$ is defined as $X_i = X_0^T \cdot B_P^i$.

By adding an additional layer, the Markov model can be extended to describe the network conditions $\mathcal{N}$ of a communication scenario $\mathcal{C}$ as concatenation of arbitrary many patterns. Given a set of M patterns $\mathcal{P} = \{P_0, \ldots, P_{M-1}\}$ and the corresponding transition matrices $\{B_{P_0}, \ldots, B_{P_{M-1}}\}$, the idea is to define a second MC $\mathcal{Y}_{\text{outer}}$ with state space $S_{\text{outer}} = \mathcal{P} = \{P_0, \ldots, P_{M-1}\}$ and nest both chains to work as an in-homogeneous Markov model. In other words, given a finite time horizon $\mathfrak{T}$, a stochastic matrix B_{outer} defining the state transitions of $\mathcal{Y}_{\text{outer}}$ and the distribution $\Lambda : \mathfrak{T} \to \mathbb{N}$ describing the length of the pattern at time $t \in \mathfrak{T}$, we can sample a sequence of data rates defined as the concatenation of M different patterns $\mathcal{P}$ by first sampling a MC of patterns $\mathcal{Y}_{\text{outer}} = (P_t)_{t\in\mathfrak{T}}$ and sampling $(\sigma_t)_{t\in\mathfrak{T}}$ sequences of length $\Lambda(t)$ from the respective T inner MCs $(\mathcal{Y}_{\text{inner}_t})_{t\in\mathfrak{T}}$ afterwards. The sequence $\Sigma = \sigma_1 \dots \sigma_T$ defined by the concatenation of sequences generated by sampling states from $(\mathcal{Y}_{\text{inner}_t})_{t\in\mathfrak{T}}$ defines the data rate changes representing the network conditions $\mathcal{N}$.

In the following, this in-homogeneous Markov model is called $Model_B(\mathfrak{T},\mathcal{P},B_{\text{outer}}, \lambda)$. To save memory space, we introduce a function $\theta : \mathfrak{T} \to \{0, \dots, M-1\}$ maps time steps $t \in \mathfrak{T}$ to patterns indices of the patterns $P_{\theta(t)} \in \mathcal{P}$, $\theta(t) \in 0, \dots, M-1$ representing the order of MC $\mathcal{Y}_{\text{outer}} = (P_t)_{t\in\mathfrak{T}}$ and thus the output of $Model_B$. As a result, we get the following definition for the network conditions $\mathcal{N}$ of a communication scenario $\mathcal{C} = (\mathcal{N}, \mathcal{U})$:

Definition 4.1.4 (Network conditions of a communication scenario). *The network conditions $\mathcal{N}$ of a communication scenario $\mathcal{C} = (\mathcal{N}, \mathcal{U})$ are defined as the seven-tuple $\mathcal{N} = (\vec{X}_0, S, \mathcal{P}, \theta, \Lambda, \mathfrak{T}, \Sigma)$, where $\vec{X}_0$ is the initial state vector, $\mathcal{P} = \{P_0, \dots, P_{M-1}\}$ is a set of M different patterns, θ represents the order of $\mathcal{Y}_{outer}$, $\Lambda : \mathcal{T} \to \mathbb{N}$ is the distribution describing the length of pattern $P_{\theta(t)} \in \mathcal{P}$, θ is the function $\theta : \mathfrak{T} \to \{0, \dots, M-1\}$ representing the order of $\mathcal{Y}_{outer}$ and $\mathfrak{T}$ is the time horizon of the in-homogeneous Markov model $Model_B(\mathfrak{T},\mathcal{P},B_{outer}, \lambda)$.*

4.1.1.1 The update and the sample function

To adapt the model to work in a communication system composed of tactical radios connected via a wireless link, we need to define an update function ϕ_{update} that considers the time interval $\lambda : \mathfrak{T} \to \mathbb{N}$ in seconds (sec) for changing the data rate defining the quality of the radio link connection.

Definition 4.1.5 (Update function). *Let $x \in [0, 1]$ be an i.i.d. random number and p_{ij} the conditional probability of state $s_i \in S$ given state $s_j \in S$ defined in transition matrix B_P, such as the exemplary matrices in (Equation 4.1). Moreover, $\lambda : \mathfrak{T} \to \mathbb{N}$ defines how long the link will stay in the next state $X_{t+1} = s_j$. Thus, given the current state $X_t = s_i$, the random number x and λ_{t+1}, the update function ϕ_{update} (Equation 4.2) computes the next state X_{t+1} by checking if x is within the probability interval of a particular state.*

$$X_{t+1} = \phi_{\text{update}}(X_t = s_i, x; B_P, \lambda_{t+1}) = \begin{cases} s_0, & \text{for } x \in [p_{i0}, p_{i1}) \\ s_1, & \text{for } x \in [p_{i1}, p_{i2}) \\ s_2, & \text{for } x \in [p_{i2}, p_{i3}) \\ s_3, & \text{for } x \in [p_{i3}, p_{i4}) \\ s_4, & \text{for } x \in [p_{i4}, p_{i5}) \\ s_5, & \text{for } x \in [p_{i5}, 1] \end{cases} \tag{4.2}$$

```
Input: $\vec{X}_t$, $\mathbf{B}_P$, seed, $\lambda_{t+1}$
Output: $\vec{X}_{t+1}$
  Initialization :
  n ← $\vec{X}$.length
  $\vec{X}_{t+1} \leftarrow \vec{0}$
  $x \leftarrow$ randomNumber(seed, min=0, max=1)
  $\vec{Y} \leftarrow \vec{X}_t * \mathbf{B}_P$
  for $i = 0$ to $n - 1$ do
      if $(x < \vec{Y}[i])$ then
          $\vec{X}_{t+1}[i] \leftarrow 1$
          break
      end if
  end for
  Set link state time to $\lambda_{t+1}$
  return $\vec{X}_{t+1}$
```

Algorithm 1 : StateUpdate [45]

For example, let us assume that the current state of the inner MC $\mathcal{Y}_{\text{inner}}$ is s_i, then s_j is chosen as next state if x is in between p_{ij} and p_{ij+1} for $j \in [0, ..., 5]$, where $p_{ij+1} = 1$ for $j = 5$. Since x is a random number between 0 and 1, and the lines of the transition matrix of a MC always sum up to 1, the process will always compute a next state X_{t+1} following the chain. Moreover, λ_{t+1} in ϕ_{update} changes according to the time distribution for state update λ. Thus λ defines how long the link will stay in a given state, which represents a particular link data rate.

An exemplary implementation of the update function is shown in Algorithm 1. For simplicity, let us introduce the N-dimensional vector representation to describe the current state $\vec{X}_t$ and next state $\vec{X}_{t+1}$ of the MC. Where $\vec{X}_t$ has value 0 for all entries up to position i which has value 1, describing that $X_t = s_i$. The same holds for $\vec{X}_{t+1}$ and $\vec{Y}$, respectively. Given the current system state $\vec{X}_t$, a matrix B_P holding the conditional probabilities for the state transitions of a particular pattern, a random *seed* and the state time parameter λ_{t+1} as input, Alg. 1 calculates the next state $\vec{X}_{t+1}$ of the MC.

The update function defined in 4.1.5 and also *StateUpdate* algorithm (Algorithm 1) can also be used to sample a sequence $\Sigma = \{\sigma_1 \dots \sigma_T\}$ of length $|\Sigma| = \sum_{t=1}^{T} \Lambda(t)$ representing the network conditions $\mathcal{N} = (\vec{X}_0, S, \mathcal{P}, \Lambda, \mathfrak{T}, \Sigma)$ of a communication scenario $\mathcal{C}$ composed of T sequences drawn from patterns, which are chosen from $\mathcal{P}$ according to function θ. This is done by calling the update function ϕ_{update} for each pattern $P_{\theta(t)} \in \mathcal{P}$ w.r.t. to the current state $\vec{X}_t$, pattern matrix $B_{\theta(t)} \in P_{\theta(t)}$, random *seed* and state time distribution parameter λ.

$$\phi_{sample}(\vec{X}_0, \mathcal{P}, \theta, \Lambda, \mathfrak{T}) \longrightarrow \Sigma = (\sigma_1, ..., \sigma_T) \tag{4.3}$$

Algorithm 2 shows how to implement the sample function in Equation 4.3 by iteratively calling update function ϕ_{update} or more precise re-using the *StateUpdate Algorithm* (Algorithm 1).

4.1.1.2 Exemplary sequences of states

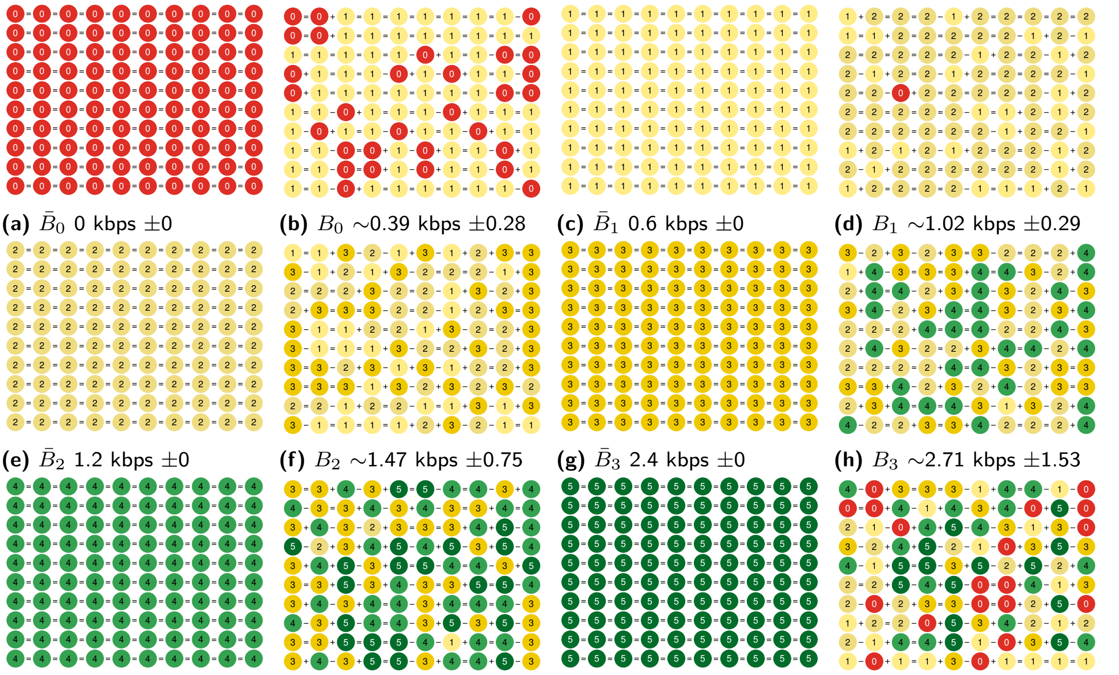
Figure 4.3 illustrates the output of the *Sample Algorithm* (Algorithm 2) for twelve different communication scenarios consisting of one single pattern. More precise, each single plot represents one single pattern out of the twelve patterns defined using the matrices in Equation 4.1, which generated the twelve sequences of states plotted in Figure 4.3. In this figure, each sequence has one hundred states out of six possible states (from 0 (red) to 5 (green)). The sequence starts from the bottom left of the plot and moves line by line from the left to the right until it reaches the top right corner. The three algebraic signs =, + and − placed in between states denote "no change", "increment" and "decrement" in the data rate, respectively. We leverage the use of these three signs and the different colors to emphasize the changes in the link data rate generated by our model.

Notice that the left column in Figure 4.3 lists stable network conditions defined by the transition matrices $\bar{B}_{0,...,5}$ in (4.1). Here, $\bar{B}_0$ represents the worst-case scenario with 0 kbps for all states (Figure 4.3a), and $\bar{B}_5$ represents the optimum scenario with all states representing maximal nominal data rate 9.6 kbps (Figure 4.3k). The right column lists the patterns of change created using $B_{0..5}$ in (4.1). The goal of these two column plots is to show a striking comparison between non-stochastic and stochastic patterns of change quantizing network conditions into discrete states.

The properties of the transition matrices can be described as follows. B_0 has high probabilities for states 0 and 1, resulting in an average data rate of ∼0.4 kbps (Figure 4.3b). B_1 has low probability of link disconnection with data rate of ∼1 kbps on average (Figure 4.3d) and B_2 is very similar to B_1, but has zero probability for link disconnection (∼1.5 kbps on average). B_3 represents a significant improvement

```
Input: $\vec{X}_0$, $\mathcal{P}$, $\theta$, $\Lambda$, $T$, $\lambda$, seed
Output: Sequence of states $\Sigma = (\sigma_1 = X_1, ..., \sigma_T = X_T)$
    Initialization :
    $\Sigma \leftarrow \{\}$
    $N \leftarrow \vec{X}_0$.length
    $\vec{X}_t \leftarrow \vec{X}_0$
    for $t = 1$ to $T$ do
        $P_m \leftarrow \mathcal{P}[\theta(t)]$
        $\mathbf{B} \leftarrow B_m$ s.t. $B_m \in P_m$
        for $l = 1$ to $\Lambda(t)$ do
            $\vec{X}_t \leftarrow StateUpdate(\vec{X}_t, \mathbf{B}, seed, \lambda_{t+1})$
            for $i = 0$ to $N - 1$ do
                if $(\vec{X}_t[i] == 1)$ then
                    $\Sigma \leftarrow \Sigma.append(\{i\})$
                end if
            end for
        end for
    end for
    return $\Sigma$
```

Algorithm 2 : Sample [45]



(a) $\bar{B}_0$ 0 kbps ±0 **(b)** B_0 ~0.39 kbps ±0.28 **(c)** $\bar{B}_1$ 0.6 kbps ±0 **(d)** B_1 ~1.02 kbps ±0.29

(e) $\bar{B}_2$ 1.2 kbps ±0 **(f)** B_2 ~1.47 kbps ±0.75 **(g)** $\bar{B}_3$ 2.4 kbps ±0 **(h)** B_3 ~2.71 kbps ±1.53

(i) $\bar{B}_4$ 4.8 kbps ±0 **(j)** B_4 ~4.85 kbps ±2.84 **(k)** $\bar{B}_5$ 9.6 kbps ±0 **(l)** B_5 ~2.84 kbps ±3.17

Figure 4.3 Six stable patterns $\bar{B}_{0..5}$ and six changing patterns $B_{0..5}$ created with Model$_B$ [45]

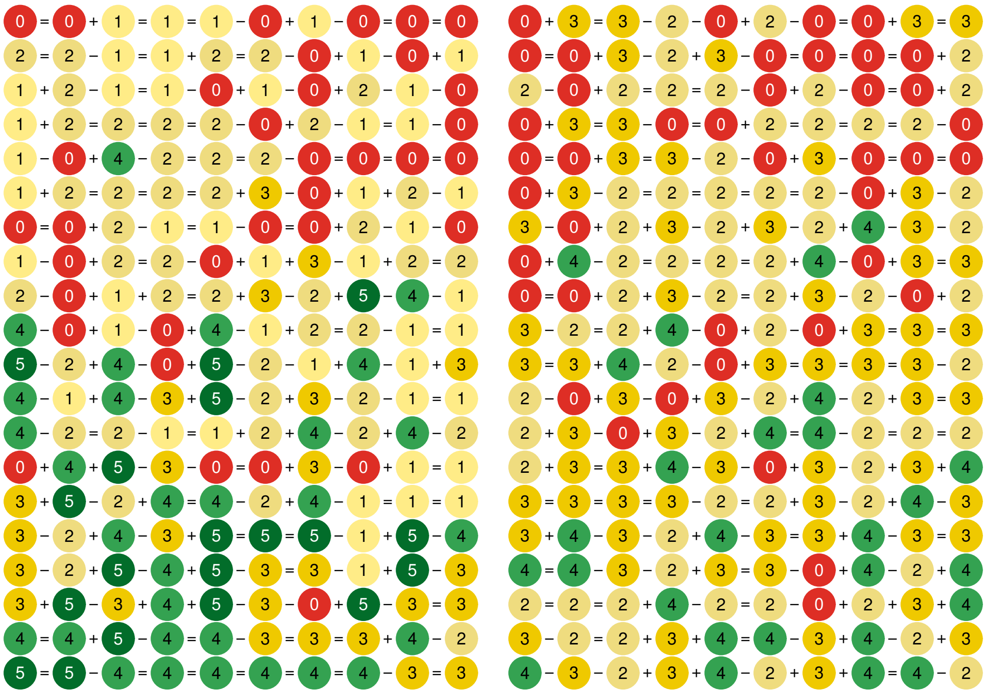
in the network conditions by moving between states 2, 3 and 4 almost surely (~2.7 kbps on average). B_4 is a slightly modification of B_3 shifting equal probabilities to the states 5, 4 and 3 resulting in 4.8 kbps (Figure 4.3j), and B_5 has equal probabilities (1/6) for all states with about 2.8 kbps on average (Figure 4.3l). These patterns create the foundation to generate more elaborated patterns of data rate change to quantify the robustness of tactical systems and thus will be re-used in the following chapters.

4.1.2 Basic morphisms among patterns

Now, let us reuse the foundations from the last section to introduce three basic procedures to morph different patterns into each other (Section 4.1.2). As referring to Section 2.3 this concept is inspired by the metamorphosis of images in the work of M.C. Escher and thus, the functions enable us to define different instances of the nested MC (Figure 4.1) to combine different patterns of change (Section 4.1.2.2) or to create loops among two or more patterns of change (Section 4.1.2.3). The first and third procedure can be interpreted as a control mechanism to “tame the randomness” in the outer MC of $Model_B$ by enforcing a specific sequence of transformations among patterns. For example, we can combine the twelve exemplary patterns in Figure 4.3 using mathematical relations. The idea behind this concept is to increase our control over both complexity and variation in these patterns of change to define experiments whose complexity also changes as a function of time.

4.1.2.1 An isomorphism among patterns

The very first morphism θ_T among patterns defines an isomorphism that transforms one pattern P_m into another pattern P_{m+1} such that $m, m+1 \in \{0, \dots, M-1\}$,



(a) T_1 ~2.34 kbps ±2.81

(b) T_2 ~3.61 kbps ±2.94

Figure 4.4 Two transformations T_1 and T_2 [45]

given the number of cross-distributions $\delta(t) > 1$ for the state transition at time $t \in \mathfrak{T}$. In the very first instance, let $\mathfrak{T} \in \mathcal{N}$ be the time horizon of a communication scenario $\mathcal{C}$ defined using $Model_B$ and assume that $\mathcal{P} = \{P_1, P_2\} \in \mathcal{N}$, meaning that the outer MC of $Model_B$ has 2 different states $S_{\text{outer}} = s_1 = P_1, s_2 = P_2$. We consider this simplified case since we can apply the transformation, described in the following paragraph, successively to each consecutive pair of patterns resulting in a generalization for communication scenarios consisting of arbitrary many patterns.

Given the number $\delta = \delta(t) > 1$ of cross-distributions $B_{T_1}, \dots, B_{T_\delta}$ for transforming P_1 into P_2 and a function $\Lambda_{\text{trans}} : \{1, \dots, \delta\} \to \mathbb{N}$ defining the length of the pattern sampled from the respective cross-distribution, the goal is to extend function θ to a function θ_T, which includes the patterns introduced by cross-distributions $B_{T_1}, \dots, B_{T_\delta}$. In the following, the patterns $P_{T_1}, \dots, P_{T_\delta}$ defined using the transformation matrices $B_{T_1}, \dots, B_{T_\delta}$ are called hidden patterns of $Model_B$, since they will be added to the description of the network conditions of the communication scenario $\mathcal{N}$ or more precise to the set of patterns $\mathcal{P} \in \mathcal{N}$, but they are hidden in the state space S_{outer} of $Model_B$. Instead, the information about the state transition is encapsulated in function θ_T, which reduces the complexity of $Model_B$ without losing any information about MC $\mathcal{Y}_{\text{outer}}$.

The idea for morphing one pattern of change into another is to increase or decrease the average link data rate incrementally. This is done by shaping the probabilities in distribution B_1 of pattern P_1 to become the probabilities of B_2 using $B_{T_1}, \dots, B_{T_{\delta-1}}$ cross-distributions. The cross-distribution matrices are computed by calculating $B_2 - B_1$ and dividing the result by the number of steps $(\delta(t) + 1 > 1)$ as defined in Equation 4.5 and vice-versa. The resulting transformation matrix B_{Trans_t} can be added $\delta(t)$ times to define the cross-distributions $P_{T_1}, \dots, P_{T_\delta}$ and transform B_1

into B_2. For example, Figure 4.4a plots the sequence of states starting with the probability distribution from B_4, which is transformed into B_0 in 8 steps ($\delta(1) = 7$, and the resulting B_{Trans_1} is shown in (4.6)); i.e. for each two lines in the figure we changed the initial matrix B_4 towards transforming it into the B_0 matrix. This process degraded the link data rate as a function of time.

Lemma 4.1.6. *Let $\theta : \mathfrak{T} \to \mathcal{P}$, $\delta : t \to \mathbb{N}$ be the number of cross-distributions for morphing $P_{\theta(t)}$ into $P_{\theta(t+1)}$, B_{Trans_t} the transformation matrix from Def. 4.1.7 and $\Lambda_T : t \to \mathbb{N}$ the distribution describing the sequence length of the transformation, then θ can be extended to a mapping including the transformation $P_{\theta(t)}$ into $P_{\theta(t+1)}$ by defining:*

$$\theta_T(t) := (t, \delta(t), B_{Trans_t}, \Lambda_T(t), B_{\theta(t)}). \tag{4.4}$$

Proof. The proof follows from the construction described in the last paragraph. □

This process can be generalized to work for arbitrary patterns being part of the network conditions $\mathcal{N} = (\vec{X}_0, S, \mathcal{P}, \theta, \Lambda, \mathfrak{T}, \Sigma)$. As mentioned before, θ is a function mapping each time step t to the index of a pattern $P_m \in \mathcal{P}$, meaning that θ represents the state transitions of the outer MC of *Model*$_B$. Using Def. 4.1.7 we can extend θ to be a mapping $\theta_T : t \cup \{0\} \to (t, \delta(t), B_{\text{Trans}_t}, \lambda_T, B_{\theta(t)})$ encapsulating the knowledge for morphing $B_{\theta(t)} \in P_{\theta(t)}$ into $B_{\theta(t+1)} \in P_{\theta(t+1)}$ in the parameters B_{Trans_t}, $\delta(t)$ and λ_T. The only requirement that must be fulfilled is that the 5-tuples $(t, \delta(t), B_{\text{Trans}_t}, \lambda_T, B_{\theta(t)})$ are well defined before creating the experiment. This can be ensured by defining the number of cross-distributions $\delta(t)$ and the number of samples per cross-distribution $\Lambda_T(t)$ and creating the transition matrices B_{Trans_t} for each point in time $t \in \mathfrak{T}$ afterwards.

Definition 4.1.7. *Let $\theta : \mathfrak{T} \to \{0, \ldots, M-1\}$ and $B_{\theta(t)} \in P_{\theta(t)}, B_{\theta(t+1)} \in P_{\theta(t+1)}$ the state transition matrices of the both patterns $P_{\theta(t)}$ and $P_{\theta(t+1)}$ respectively. Then the pattern transition matrix B_{Trans_t} that can be added $\delta(t) > 0$ times to define the cross-distribution $P_{T_1}, \ldots, P_{T_\delta}$ morphing $B_{\theta(t)}$ into $B_{\theta(t+1)}$ is defined by:*

$$B_{Trans_t} = \frac{B_{\theta(t+1)} - B_{\theta(t)}}{\delta(t) + 1} \tag{4.5}$$

More precisely, one can generate the set of cross-distributions $B_{T_1}, \ldots, B_{T_\delta}$ transforming $B_{\theta(t)} \in P_{\theta(t)}$ into $B_{\theta(t+1)} \in P_{\theta(t+1)}$ by adding B_{Trans_t} $\delta(t) + 1$ times to $B_{\theta(t)}$. Assuming that Σ is the sequence of data rate changes of $\mathcal{N}$ up to time t, we can concatenate Σ with the sequences $\sigma_{T_1}, \ldots, \sigma_{T_\delta}$ sampled from $B_{T_1}, \ldots, B_{T_\delta}$ to update the sequence representing the network conditions $\mathcal{N}$. In the case, that no cross-distributions are desired we can reduce the model to behave like its initial version by setting $\delta(t) = 0$ and $B_{\text{Trans}_t} = B_{\theta(t+1)} - B_{\theta(t)}$. Note, that we assume that $\Lambda_T(t)$ always includes the sample size of the target pattern and that $(0, 0, \mathbb{0}, \lambda(1), B_{\theta(1)})$ defines the first transformation with $\Lambda_T(0) = \Lambda(1)$, meaning that we sample exactly $\Lambda(1)$ states from the first pattern $P_{\theta(1)}$, without any transformation.

This transformation process can also be interpreted as function ϕ_T extending the communication scenario to include the morphisms of P_t into P_{t+1} defined by θ_T for each point $t \in \mathfrak{T}$. Algorithm 3 shows an exemplary implementation of the transformation function ϕ_T assuming that we can access the patterns in $\mathcal{P}$ and their components using a *key-value* data structure.

$$B_{T_1}\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & -.08375 & .08375 & 0 & 0 & 0 & 0 \\ 1 & .04125 & .0825 & -.00125 & -.04 & -.04125 & -.04125 \\ 2 & .04125 & .0825 & -.00125 & -.04 & -.04125 & -.04125 \\ 3 & .04125 & .0825 & -.00125 & -.04 & -.04125 & -.04125 \\ 4 & .04125 & .0825 & -.00125 & -.04 & -.04125 & -.04125 \\ 5 & .04125 & .0825 & -.00125 & -.04 & -.04125 & -.04125 \end{array} \quad B_{T_2}\begin{array}{c|cccccc|} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & +1 & 0 & 0 & 0 & -1 & 0 \\ 1 & +1 & 0 & 0 & 0 & -1 & 0 \\ 2 & +1 & 0 & 0 & 0 & -1 & 0 \\ 3 & +1 & 0 & 0 & 0 & -1 & 0 \\ 4 & +1 & 0 & 0 & 0 & -1 & 0 \\ 5 & +1 & 0 & 0 & 0 & -1 & 0 \end{array} \tag{4.6}$$

Definition 4.1.8. *Let $\mathcal{C} = (\vec{X}_0, S, \mathcal{P}, \theta, \Lambda, \mathfrak{T}, \Sigma)$ be a communication scenario and $\theta_T : t \to (t, \delta(t), B_{Trans_t}, \Lambda_T(t), B_{\theta(t)})$ the extension of the transformation function θ as referring to Definition 4.1.7. Then*

$$\phi_T : (\vec{X}_0, S, \mathcal{P}, \theta, \Lambda, \mathfrak{T}, \Sigma) \longrightarrow (\vec{X}_0, S, \mathcal{P}, \theta_T, \Lambda_T, \mathfrak{T}, \Sigma) \tag{4.7}$$

transforms a communication scenario to include transformations defined by θ_T.

Definition 4.1.9 (Communication scenario including transformations)**.** *We call a communication scenario $\mathcal{C}$ a "communication scenario including transformations", if $\theta \in \mathcal{N}$ is extended to θ_T and thus $\mathcal{P} = \{(t, \delta(t)), B_T, \Lambda_T(t), B_{\theta(t)}\}_{t\in\mathfrak{T}}$.*

Once θ is extended to θ_T and the set of patterns is defined as in Definition 4.1.9, the communication scenario is called a "communication scenario including transformations". Again, one can sample a sequence of patterns $\Sigma = \{\sigma_1 = X_1, ..., \sigma_T = X_T\}$, characterizing the network conditions of an experiment including transformations, by using the sample and update functions ϕ_{sample} and ϕ_{update} respectively. This is done by slightly changing the *SampleAlgorithm* (Algorithm 2) to work with input parameters $(\vec{X}_0, \mathcal{P}, \theta_T, \Lambda_T, \lambda, seed)$.

```
Input: $\mathcal{P}$, $\theta$, $\delta$,$\Lambda_T$,$\mathfrak{T}$
Output: $\mathcal{P}$, $\theta_T$
  Initialization :
$\mathcal{B} \leftarrow \{B_{\theta(1)}, \dots, B_{\theta(T)}\}$
for t $\in \mathfrak{T}$ do
    $\mathbf{B}_{\text{Trans}_t} \leftarrow (\mathcal{B}[t+1] - \mathcal{B}[t]) \,/\, (\delta(t) + 1)$
    if $(\delta(t) > 1)$ then
        for (j = 1 to $\delta(t)$) do
            $\theta_T(t) \leftarrow (t, \delta(t), \mathbf{B}_{\text{Trans}_t}, \Lambda_T(t), \mathcal{B}[t])$
            $\mathcal{P}_T \leftarrow \mathcal{P}_T \cup \{(t, \delta(t), \mathbf{B}_{\text{Trans}_t}, \Lambda_T(t), \mathcal{B}[t])\}$
        end for
    else
        $\theta_T(t) \leftarrow (t, 0, (\mathcal{B}[t+1] - \mathcal{B}[t]), \Lambda_T(t), \mathcal{B}[t])$
        $\mathcal{P}_T \leftarrow \mathcal{P}_T \cup \{(t, 0, (\mathcal{B}[t+1] - \mathcal{B}[t]), \Lambda_T(t), \mathcal{B}[t])\}$
    end if
end for
$\mathcal{P} \leftarrow \mathcal{P}_T$
return $\mathcal{P}$, $\theta_T$
```

Algorithm 3 : PatternTransformation [45]

Another way of morphing one pattern into another is to use a predefined transformation matrix, like B_{T_2} in (4.6), to transform the underlying probability distribution of one pattern to become the distribution of the second one. The only restriction on the transformation matrix is that it is symmetric, meaning that the result after each application is a stochastic matrix again (the lines sum up to one). In this case, $\delta(t)$ can be interpreted as scaling factor controlling the speed of the transformation. More precisely, we divide B_{Trans_t} by $\delta(t)$ as defined in Equation 4.8. Thus, $B_{\theta(t+1)}$ represents the transformed matrix at time $t+1$, $B_{\theta(t)}$ is the probability distribution matrix at time t, B_{Trans_t} is the predefined transformation matrix, and $\delta(t)$ defines the number of cross-distributions used to morph $B_{\theta(t)}$ into $B_{\theta(t+1)}$. An example scenario, illustrating how the matrix B_{T_2} (Equation 4.8) can be used to increase the probability of state 0 and decrease the probability of state 4 is shown in Figure 4.4b. In this figure, we observe several appearances of state 4 (light green) in the beginning (bottom rows), which are incrementally replaced by state 0 (red), while moving forward in time, therefore, also degrading the link data rate as a function of time.

$$B_{\theta(t+1)} = B_{\theta(t)} + \frac{B_{\mathrm{Trans}_t}}{\delta(t)+1} \tag{4.8}$$

4.1.2.2 Jumps among patterns

The above section demonstrated how to gain control of the outer MC by using functions θ and θ_T morphing one pattern into another. Another way of generating the network conditions is to trade control of the model for complexity of the generated network conditions by *"jumping"* from one probability distribution into another. In this case, the model follows the distribution of the transition matrix B_{outer} coupled with the distributions of the patterns defining the inner MC. The resulting model $Model_B(\mathfrak{T},\mathcal{P},B_{\mathrm{outer}},\ \lambda)$ is illustrated in Figure 4.1. In this case, the first pattern defining state zero follows the distribution of $\bar{B}_0$ (static pattern), the next one follows the distribution of B_1 and so on until state five defining the pattern distributed by $\bar{B}_5$; these transition matrices were previously defined in (Equation 4.1) and plotted in Figure 4.3. Therefore, the probability distributions in (Equation 4.1) are used by the update function ϕ_{update} as defined in Equation 4.9. For example, given that the outer MC is in state $X_t = s_i$ and two random numbers $x_1, x_2 \in [0,1]$, the next state $X_{t+1} = s_j$ is computed by determining the next step transition matrix $B_{\theta(t+1)} \in P_{\theta(t+1)}$ matching x_2 and the probability intervals from the B_{outer} distribution and calling the update function ϕ_{update} with sampling parameter x_1, transition matrix $B_{\theta(t+1)}$ and time distribution λ_{t+1} afterwards. This is called in-homogeneous MC, because the probability distribution may change as a function of time (i.e. $t \in \mathfrak{T} = 1, \ldots, T$) [29].

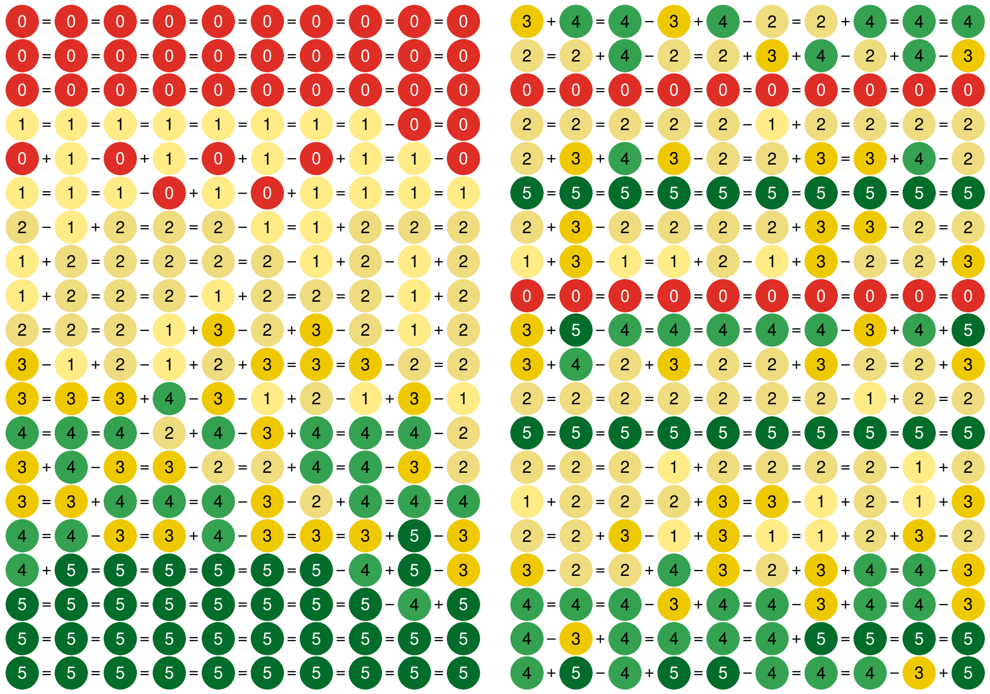
$$\phi_{\text{outer}}(s_i, x_1, x_2; B_{outer}, \lambda) = \begin{cases} \phi_{\text{update}}(s_i, x_1, B_1, \lambda_{t+1}), & \text{for } x_2 \in [p_{i0}, p_{i1}) \\ \vdots & \\ \phi_{\text{update}}(s_i, x_1, B_m, \lambda_{t+1}), & \text{for } x_2 \in [p_{im-1}, p_{im}) \\ \vdots & \\ \phi_{\text{update}}(s_i, x_1, B_M, \lambda_{t+1}), & \text{for } x_2 \in [p_{iM}, 1] \end{cases} \tag{4.9}$$

Given a set of patterns $\mathcal{P} = P_1, \dots, P_M$ and the corresponding transition matrices $B_1, \dots, B_M$, an initial state vector $\vec{X_0}$ and time distribution $\lambda \subseteq \mathbb{N}$ the sequence of states $\Sigma = \{\sigma_1, \dots, \sigma_T\}$ over time $\mathfrak{T} = \{1, \dots, T\}$ following the in-homogeneous model $Model_B(B,\ \mathfrak{T},\ \lambda)$ can be computed by first sampling $|T|$ transition matrices using distribution B_{outer} and applying the *Sample algorithm* (Alg. 2) afterwards. To this end, let $B_m \in P_m$ be the transition matrix sampled at time t, then $\theta(t)$ is set to m guaranteeing that θ is a well defined mapping $\mathfrak{T} \to \{0, \dots, M-1\}$ after the sampling process.

Moreover, we define $\delta : \mathfrak{T} \to \{1\}$ to be the simple mapping $\delta(t) = 1\ \forall t \in \mathfrak{T}$. That is, the process will change the state distribution in one step that we call "jump". Now, we can apply *Sample algorithm* (Alg. 2) with input ($\vec{X_0}$,$\mathcal{P}$,θ,Λ, T, λ) and a random *seed* to get a sequence Σ representing "jumps" among patterns $\mathfrak{B} = B_1, \dots, B_M$

```
Input: $\vec{X_0}$, $\mathcal{P}$, $\Lambda$, $\mathbf{B}_{\text{outer}}$, $\theta$, $T$, $\lambda$, seed
Output: Sequence of states $\Sigma = \{\sigma_1 = X_1, ..., \sigma_T = X_T\}$
  Initialization :
  $\mathfrak{B} \leftarrow \{B_1, \dots, B_m, \dots, B_M\}$ s.t. $B_m \in P_m \in \mathcal{P}$
  $M \leftarrow |\mathfrak{B}|$
  $m \leftarrow$ randomNumber(seed, min=1, max=$M$)
  $B_t \leftarrow \mathfrak{B}[m]$
  $\vec{X}_{inh} \leftarrow \vec{0}$
  $\vec{X}_{inh}[m] \leftarrow 1$
  $\theta(1) \leftarrow m$
  for $t = 1$ to $T$ do
    $x \leftarrow$ randomNumber(seed, min=0, max=1)
    $\vec{X}_{inh} \leftarrow \vec{X}_{inh} * \mathbf{B}_{\text{outer}}$
    for $i = 0$ to $M - 1$ do
      if ($\vec{X}_{inh}[i] == 1$) then
        $\theta(t) \leftarrow i$
      end if
    end for
    $\delta(t) \leftarrow 1$
  end for
  $\Sigma \leftarrow Sample(\vec{X_0}, \mathcal{P}, \theta, \Lambda, T, \lambda, seed)$
  return $\Sigma$
```

Algorithm 4 : Jump [45]



(a) J_1 ~2.98 kbps ±3.46

(b) J_2 ~3.19 kbps ±3.09

Figure 4.5 Two jumps J_1 and J_2 [45]

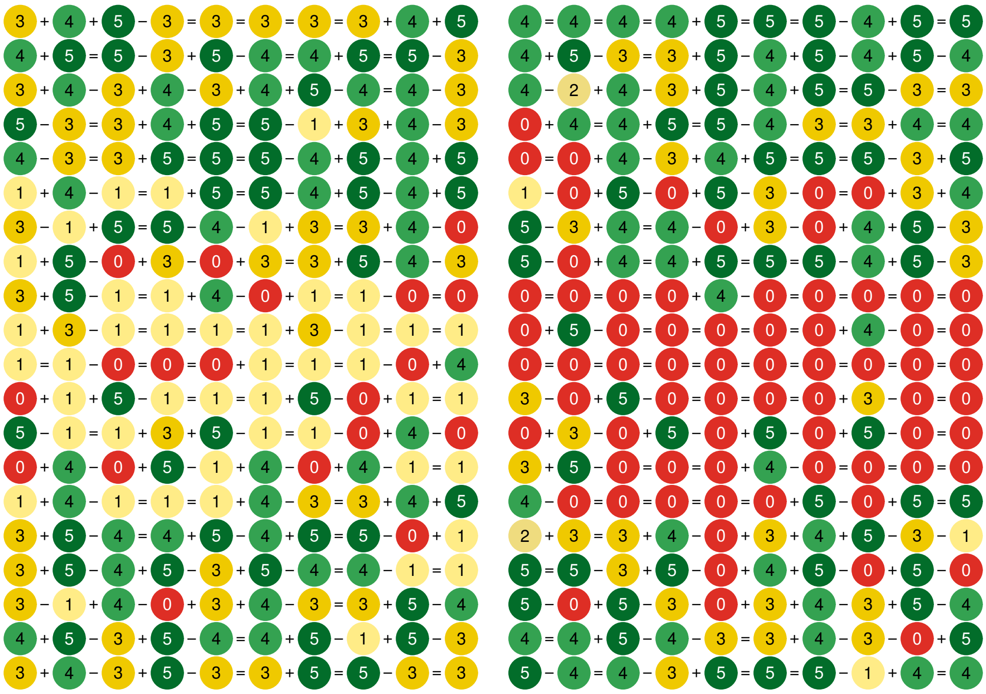
defined in ϕ_{outer} (Equation 4.9). For the sake of completeness, however, it should be noted that we could also generate "smooth jumps" among patterns by applying ϕ_T (Alg. 3) before we use the *Sample algorithm* (Alg. 2).

An algorithm returning a sequence Σ distributed according to the nested MC shown in 4.1 is given in Alg. 4. Using an initial vector $\vec{X}_0$, a set of patterns $\mathcal{P}$ with length Λ, the transition matrix for the outer MC B_{outer}, a place holder for function θ, the maximum number of time steps T of the experiment, a time distribution λ for the step length of the system states and a random *seed* as input, these algorithm computes the network changes of a communication scenario "jumping" between distributions. Using this inputs the algorithm computes Σ by first sampling T transition matrices, updating θ and calling the *Sample algorithm* (Alg. 2) as described before.

Figure 4.5a illustrates the output of the *"jump"* function, plotting a sequence of states J_1 starting with the best conditions in $\bar{B}_{5(\text{green})}$ and goes towards $\bar{B}_{0(\text{red})}$ jumping (every three lines) through B_4, B_3, B_2 and B_1 in this sequence. Thus, using a deterministic way of choosing stochastic patterns. This model can also create extreme variations in the link data rate by using another MC to define the probability distribution for state update. For example, using equal probabilities among the six states (say from B_5 in (4.1)) one can create the sequence of states in Figure 4.5b, where each line was generated by a different probability distribution; let us call it J_2 to differentiate from the first example J_1.

4.1.2.3 Loops among patterns

As explained in Section 2.3 the stochastic model and also the functions for transforming patterns into each other are inspired by the art of M.C. Escher and in particular



(a) L_1 ~3.19 kbps ±3.09 **(b)** L_2 ~3.87 kbps ±3.80

Figure 4.6 Two loops L_1 and L_2 [45]

by his metamorphosis work. In the case of modeling network scenarios, the functions ϕ_T and ϕ_{sample} are the basic foundations to create *loops* of patterns morphing one probability distribution into another one. The construction of such a metamorphosis goes as follows. Assuming that the start and end pattern $\mathcal{P} = \{P_0, P_1\}$, the function for the pattern length Λ_T and the function $\delta : \mathfrak{T} \to \mathbb{N}$ for the number of cross distributions of the metamorphosis is given, one can call the *Pattern Transformation* algorithm (6) to transform P_0 to P_1 (in $\delta(1)$ steps) and vice versa ($\delta(2)$ steps). Finally the sequence of data rates can be computed using the *Sampling Algorithm* (Alg. 2).

As a result, the morphism starts with the first pattern and subsequently transforms one cross-distribution into the next until the transformation stops at the target pattern. Then, it gets back to the initial pattern by inverting the whole process. Moreover, we can create loops using "jumps" earlier defined in Section 4.1.2.2, but in this case we observe abrupt changes in the link data rate as illustrated by J_2 in Figure 4.5b. This is because each transition morphing one probability distribution into the next one is done in one single step.

Two exemplary loop scenarios with 200 states are shown in Figure 4.6. The first one morphing from B_4 to B_1 and back to B_4 and the second one replacing B_1 by $\bar{B}_0$. Comparing both scenarios, we observe that replacing B_1 by $\bar{B}_0$ results in more frequent link disconnections in the middle of the scenario. We will discuss the effects of these series of frequent disconnections (red area in the plot) on the performance of the TS later in Section 5.

4.2 *Model*$_{B^*}$: Enhancing the stochastic *Model*$_B$

The following section enhances *Model*$_B$ to a multi-agent model, called *Model*$_{B^*}$. Thus, the enhancement is broken down into two steps: First, we provide an enhancement of *Model*$_B$ based on the theory of OT, thus facilitating smoother pattern transitions by finding the most efficient way to move the mass distribution of one pattern into another. Moreover, this enhancement enables us to transform the Markov model *Model*$_{B^*}$ to a MDP to automate the experiment generation process. Second, we introduce the multi-agent *Model*$_{B^*}$ composed of two agents, one solving the MDP to create link data rate changes defining the network conditions of the tactical system and a second intelligent agent learning the distribution of the TtR after unplanned link disconnections, thus introducing a first metric to measure the robustness of the system in the worst-case scenario.

4.2.1 Optimal transport and Wasserstein transformation

The motivation for enhancing *Model*$_B$ based on OT comes from the fact that from solution of the OT problem we get two main aspects: First, defining and solving the OT introduces a metric to the underlying problem space. This metric, also called Wasserstein distance, measures the similarity between different patterns even for smallest perturbations in the respective distributions. Second, the optimal mapping (OT matrix, Monge mapping) describes the correspondence between arbitrary many patterns. Combining both, OT enables us to transform one pattern of data rate into another by moving a minimum amount of mass with respect to the underlying probability distribution. Exploiting this fact, we can use barycentric interpolation to generate a set of cross-distributions fulfilling a nearest neighbour property with respect to the Wasserstein distance and thus defining the patterns smoothly morphing one pattern of data rate change into another one. This can be seen as a more natural way than using the transformation function in Section 4.1 and offers the advantage that we can analyze the changes in the network while morphing two patterns at arbitrary scale. One exemplary approach making this happen is shown in Section . Then, Section 4.2.2 shows how we can apply this approach to define a multi-agent model to measure the robustness of tactical systems by computing the distribution of the TtR IP data flows after unplanned link disconnections.

4.2.1.1 Theoretical foundations

The following section describes the theoretical foundations and also the application of OT to solve *Problem B*. This theoretical digression follows Chapter 2 of [61] and will be applied to enhance *Model*$_B$ afterwards. Since the theory of OT is well studied and also the notations are generalized, we do not claim any rights on the contents of this section.

Let us start, defining the probability simplex Δ_n for the histogram or probability vector $a \in \Delta_n$ as

$$\Delta_n := \left\{ a \in \mathbb{R} : \sum_{i=1}^{n} a_i = 1 \right\}. \tag{4.10}$$

Since *Model$_B$* is defined as a discrete MC, created to generate ever-changing sequences of data rates, this theoretical part focuses on OT for discrete measures. Nonetheless, it can also be expanded to the continuous case. So let δ_x be the Dirac at position x, then a discrete measure with weights $\mathbf{a}$ and locations $x_1, \dots, x_n \in \mathcal{X}$ on space $\mathcal{X}$ reads

$$\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i}. \tag{4.11}$$

Moreover, force $\mathbf{a} \in \Delta_n$ and $\mathbf{a} > 0$, meaning that Equation 4.11 defines a positive probability measure. Now, we can define the *optimal assignment problem* and, based on this, the *Monge problem* between discrete measures. To this end, let $(\mathbf{C}_{i,j})_{i\in[n],j\in[n]}$ be a cost matrix, then the goal of the optimal assignment problem is to find a bijection $\hat{p}$ fulfilling

$$\min_{\hat{p}\in Perm(n)} \frac{1}{n} \sum_{i=1}^{n} \mathbf{C}_{i,\hat{p}(i)}, \tag{4.12}$$

where Perm(n) is the set of permutations over n elements. Now, the idea of the *Monge transportation problem* [55] for discrete measures is to find a map that pushes the mass of α from point x_i towards the mass of β at point y_j. More formal, the goal is to find a map $T : \{x_1, \dots, x_n\} \to \{y_1, \dots, y_m\}$ s.t.

$$\forall j \in [m], \mathbf{b}_j = \sum_{i:T(x_i)=y_j} a_i = \beta = T_{\#}\alpha. \tag{4.13}$$

Moreover, the mapping has to fulfill the condition to minimize the transportation cost as referring to the assignment problem $c(x, y)$ defined on $(x, y) \in X \times Y$

$$\min_{T} \left\{ \sum_{i} c(x_i, T(x_i)) : T_{\#} = \beta \right\}. \tag{4.14}$$

If the values for x and y are distinct, we can encode the the Monge map by defining the permutation operator $\hat{p} : [n] \to [m]$ s.t. $j = \hat{p}(i)$ and the mass conservation by

$$\sum_{i\in\hat{p}^{-1}(j)} \mathbf{a}_i = \mathbf{b}_j. \tag{4.15}$$

The assignment problem in Equation 4.12 can be seen as special case of the Monge problem with cost matrix $\mathbf{C}_{i,j} := c(x_i, y_j)$, when $n = m$ and $\mathbf{a}_i = \mathbf{b}_j = \frac{1}{n}$. This is, because Equation 4.15 implies that T is a bijection fulfilling $T(x_i) = y_{\hat{p}(i)}$.

Now, the point is that in practice both problems have its drawbacks. The assignment problem with its formulation as permutation problem can only be used to compare uniform histograms fulfilling $n = m$. For the generalized Monge problem (4.13), the formulation can be degenerate meaning that there is no feasible solution satisfying Equation 4.15. Moreover, since the assignment problem is combinatorial and the

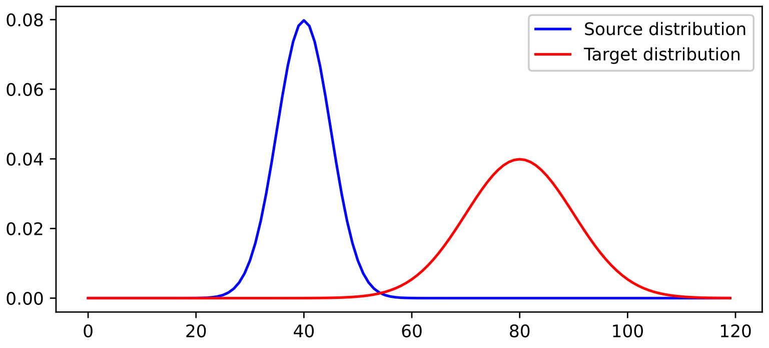


Figure 4.7 Two gaussian distributions

set of feasible solutions of the Monge problem is non-convex, both problems are computational cost intensive and difficult to solve.

For this reason, Kantorovich [33] introduced a relaxed formulation of the Monge problem. To this end, he weakened the condition that one point x_i can be assigned to one point $y_{\hat{p}}$ or $T(x_i)$ only by allowing that the mass at point x_i can be distributed across multiple locations of the target distribution. The idea is that mass transportation becomes probabilistic by splitting the mass over several targets, instead of being deterministic. To do so, Kantorovich replaces the permutation $\hat{p}$ or map T by a matrix $\mathbf{P} \in \mathbb{R}_+^{n \times m}$ describing the mass moving from x_i to y_j. One can define the set of admissible coupling matrices $\mathcal{U}(\mathbf{a}, \mathbf{b})$ by

$$\mathcal{U}(\mathbf{a}, \mathbf{b}) := \left\{ \mathbf{P} \in \mathbb{R}_+^{n \times m} : \mathbf{P}\mathbb{1}_m = a \text{ and } \mathbf{P}^T \mathbb{1}_n \right\}. \tag{4.16}$$

This set is bounded, because it has to fulfill $n + m$ equality constraints and as a result it is a convex polytope comparing to the Monge problem, where the set of feasible solution was non-convex. Moreover, the Kantorovich formulation is always symmetric, because a coupling matrix $\mathbf{P}$ is in $\mathbf{U}(\mathbf{a}, \mathbf{b})$ if and only if $\mathbf{P}^T$ is in $\mathbf{U}(\mathbf{b}, \mathbf{a})$. Following this ideas Kantorovich's optimal transport problem (OT) is to find a coupling matrix $\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})$ such that

$$\mathcal{L}_{\mathbf{C}}(\mathbf{a}, \mathbf{b}) := \min_{\mathbf{P} \in \mathcal{U}(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \mathbf{P} \rangle := \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j}, \tag{4.17}$$

which is a linear program, meaning that its optimal solutions are not necessarily unique.

Now, the Kantorovich relaxation can be applied to discrete measures α, β as introduced in Equation 4.13. To this end, we define the matrix $\mathbf{C}$ to hold the pairwise costs between the locations x_i and y_j, namely $\mathbf{C}_{i,j} := c(x_i, y_j)$ and

$$\mathcal{L}_C(\alpha, \beta) := L_{\mathbf{C}}(\mathbf{a}, \mathbf{b}). \tag{4.18}$$

This is what we refer to as the *Kantorovich formulation* of optimal transport between discrete measures. The problem can also be reinterpreted using random variables meaning that Problem 4.18 is equivalent to

$$\mathcal{L}_C(\alpha, \beta) := \min_{(X,Y)} \left\{ \mathbb{E}_{(X,Y)}(c(X,Y)) : X \sim \alpha, Y \sim \beta \right\} \tag{4.19}$$

In the probabilistic interpretation of the problem the tuple (X, Y) is defined over the space $\mathcal{X} \times \mathcal{Y}$. Moreover, the law of X must be α and the law of Y must be β respectively. As a result, the law of the tuple (X, Y) then is $\pi \in U(\alpha, \beta) \in \mathcal{X} \times \mathcal{Y}$.

4.2.1.2 Metric properties of optimal transport (OT)

One of the most important properties of OT is that it "lifts" a distance between bins to a metric between histograms on that bins, given that the matrix $\mathbf{C}$ fulfills some constraints. In the simplest case the matrix $\mathbf{C}$ is fixed, meaning that it holds the pairwise costs between bins to be compared. Following [68], it can be shown that OT introduces a distance between the histograms defined on that bins.

Proposition 4.2.1 (Wasserstein distance between discrete measures [61])**.** *Suppose* $n = m$ *and for some* $p \geq 1$, $\mathbf{C} = \mathbf{D}^p = \left(\mathbf{D}^p_{i,j}\right)_{i,j} \in \mathbb{R}^{n \times n}_+$, *s.t.* $\mathbf{D} \in \mathbb{R}^{n \times n}_+$ *is a distance on* $[n]$, *i.e.*

(i) $\mathbb{R}^{n \times n}_+$ *is symmetric;*

(ii) $\mathbf{D}_{i,j} = 0$ *if and only if* $i = j$*;*

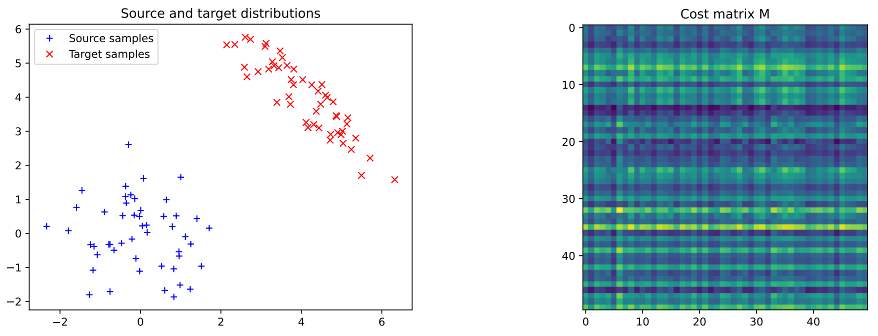
(iii) $\forall\, (i, j, k) \in [n]^3, \mathbf{D}_{i,k} \leq \mathbf{D}_{i,j} + \mathbf{D}_{j,k}$

Then

$$\mathcal{W}_p(\mathbf{a}, \mathbf{b}) := L_{\mathbf{D}^p}(\mathbf{a}, \mathbf{b})^{\frac{1}{p}} \tag{4.20}$$

defines the p*-Wasserstein distance on* Δ_n*, meaning that it fulfills conditions* (i)*-*(iii)*.*

Proof. A detailed proof can be found in [61]. □



(a) Empirical source and target distribution **(b)** OT cost matrix

Figure 4.8 2D optimal transport problem between two empirical distributions

Moreover Prop. 4.2.1 can be generalized to work for arbitrary measures

Proposition 4.2.2. *Assume $\mathcal{X} = \mathcal{Y}$ and that there exists $p \geq 1$ s.t. $c(x,y) = d(x,y)^p$. Here, d is a distance defined on $\mathcal{X}$, i.e.*

(i) $d(x,y) = d(y,x) \geq 0$;

(ii) $d(x,y) = 0$ *iff* $x = y$;

(iii) $\forall (x,y,z) \in \mathcal{X}^3, d(x,z) \leq d(x,y) + d(y,z)$.

Then the p-Wasserstein distance on $\mathcal{X}$ is defined as

$$\mathcal{W}_p(\alpha,\beta) := \mathcal{L}_{d^p}(\alpha,\beta)^{\frac{1}{p}}. \tag{4.21}$$

Since $\mathcal{W}_p(\alpha,\beta)$ is a distance it is symmetric, non-negative, $\mathcal{W}_p(\alpha,\beta) = 0$ iff $\alpha = \beta$ and $\mathcal{W}_p$ satisfies the triangle inequality

$$\forall (\alpha,\beta,\gamma) \in \mathcal{M}_+^1(\mathcal{X})^3, \mathcal{W}_p(\alpha,\gamma) \leq \mathcal{W}_p(\alpha,\beta) + \mathcal{W}_p(\beta,\gamma). \quad [61] \tag{4.22}$$

It should be noted, that the Wasserstein distance has many significant properties. For this study, the most important is that $\mathcal{W}_p$ is a weak distance meaning that it allows to compare singular distributions as discrete ones. More formal this means that for any $p > 0$, $\mathcal{W}_p^p(\delta_x, \delta_y) = d(x,y)$. Another consequence is that we have $\mathcal{U}(\delta_x,\delta_y) = \{\delta_{x,y}\}$ and thus the Kantorovich problem in Equation 4.17 has only one feasible solution $\mathcal{W}_p^p(\delta_x,\delta_y) = (d(x,y)^p)^{\frac{1}{p}} = d(x,y)$. As a result, $\mathcal{W}_p$ fulfills $\mathcal{W}_p(\delta_x,\delta_y) \to 0$ if $x \to y$, which means that $\mathcal{W}_p$ can be used to quantify the weak convergence (Def. 4.2.3).

Definition 4.2.3 (Definition of Weak Convergence [61])**.** *Let $\mathcal{X}$ be a compact domain, then a sequence $(\alpha_k)_k$ converges weakly to α in $\mathcal{M}_+^1(\mathcal{X})$, $\alpha_k \rightharpoonup \alpha$ iff $\int_{\mathcal{X}} g d\alpha_k \to \int_{\mathcal{X}} g d\alpha$ holds for any continuous function $g \in \mathfrak{C}(\mathcal{X})$.*

4.2.1.3 Entropic regularization and Sinkhorn algorithm

This section introduces numerical schemes to solve the Kantorovich problem stated in Equation 4.18 approximately. The idea is to add an entropic regularization penalty to the Kantorovich problem, which allows to find an optimal solution for the regularized problem by using an alternate minimization scheme, which consists of simple matrix-vector products.

To this end, let $\mathbf{P}$ be a coupling matrix of the optimal transport problem 4.17. The discrete entropy of $\mathbf{P}$ can be defined as

$$\mathbf{H}(\mathbf{P}) := \sum_{i,j} \mathbf{P}_{i,j} \left(\log(\mathbf{P}_{i,j}) - 1\right). \tag{4.23}$$

Moreover, define $\mathbf{H}(\mathbf{a}) = -\infty$ if there exist an entry $\mathbf{a}_j$ of $\mathbf{P}$ with $\mathbf{a}_j \leq 0$. Since $\partial^2 \mathbf{H}(\mathbf{P}) = -\text{diag}(\frac{1}{\mathbf{P}_{i,j}})$ and $\mathbf{P}_{i,j} \leq 1$, the entropy function is strongly concave. Now, one can add the negative of $\mathbf{H}$ as regularization term to the Kantorovich formulation of optimal transport to get approximate solutions for the original optimal transport problem in Equation 4.18:

$$L^{\varepsilon}_{\mathbf{C}}(\mathbf{a},\mathbf{b}) := \min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \langle \mathbf{P},\mathbf{C}\rangle - \varepsilon\mathbf{H}(\mathbf{P}). \tag{4.24}$$

This problem has a unique solution, because the objective function $L^{\varepsilon}_{\mathbf{C}}(\mathbf{a},\mathbf{b})$ is ε-strongly convex. The entropic regularization term can geometrically be interpreted as pushing the original solution away from the boundary toward an entropic center of the convex polygon defining the problem space.

Proposition 4.2.4 (ε-Convergence [61])**.** *Let $\mathbf{P}_\varepsilon$ be the unique solution of Equation 4.24. Then, $\mathbf{P}_\varepsilon$ converges to the optimal solution of Kantorovich problem with maximal entropy or more formal*

$$\mathbf{P}_\varepsilon \xrightarrow{\varepsilon\to 0} \arg\min_{\mathbf{P}} \left\{-\mathbf{H}(\mathbf{P}) : \mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b}), \langle\mathbf{P},\mathbf{C}\rangle = L_{\mathbf{C}}(\mathbf{a},\mathbf{b})\right\}, \tag{4.25}$$

s.t.

$$L^{\varepsilon}_{\mathbf{C}}(\mathbf{a},\mathbf{b}) \xrightarrow{\varepsilon\to 0} L_{\mathbf{C}}(\mathbf{a},\mathbf{b}). \tag{4.26}$$

Furthermore,

$$\mathbf{P}_\varepsilon \xrightarrow{\varepsilon\to\infty} \mathbf{a}\otimes\mathbf{b} = \mathbf{a}\mathbf{b}^T = (\mathbf{a}_i\mathbf{b}_j)_{i,j}. \tag{4.27}$$

Proposition 4.2.4 can be interpreted as follows. Equation 4.26 shows that for small regularization ε, the solution P_ε converges to the maximum entropy coupling of optimal transport. In contrast, Equation 4.27 means that as ε increases, the optimal coupling matrix becomes less sparse. This leads to faster convergence and less computational cost. As a result, choosing the appropriate value for ε is an important task when solving the optimal transport problem.

Now, the unique solution of Equation 4.24 can be projected onto $\mathbf{U}(\mathbf{a},\mathbf{b})$ of the Gibbs kernel associated to the cost matrix $\mathbf{C}$. An example showing the cost matrix for two empirical distributions (Figure 4.8a) can be seen in 4.8b. To do so, define the Kullback-Leibler distance between any two couplings as

$$\mathbf{KL}(\mathbf{P},\mathbf{K}) := \sum_{i,j} \mathbf{P}_{i,j}\log\left(\frac{\mathbf{P}_{i,j}}{\mathbf{K}_{i,j}}\right) - \mathbf{P}_{i,j} + \mathbf{K}_{i,j} \tag{4.28}$$

and

$$\mathbf{K}_{i,j} := \exp\left(-\frac{\mathbf{C_{i,j}}}{\varepsilon}\right) \tag{4.29}$$

.

As a consequence, by plugging $\mathbf{K}_{i,j}$ in Equation 4.28 one has

$$\mathbf{P}_\varepsilon = \mathrm{Proj}^{\mathbf{KL}}_{\mathbf{U}(\mathbf{a},\mathbf{b})}((K)) = \arg\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \mathbf{KL}(\mathbf{P}|\mathbf{K}) \tag{4.30}$$

for the unique optimal solution of regularized transport.

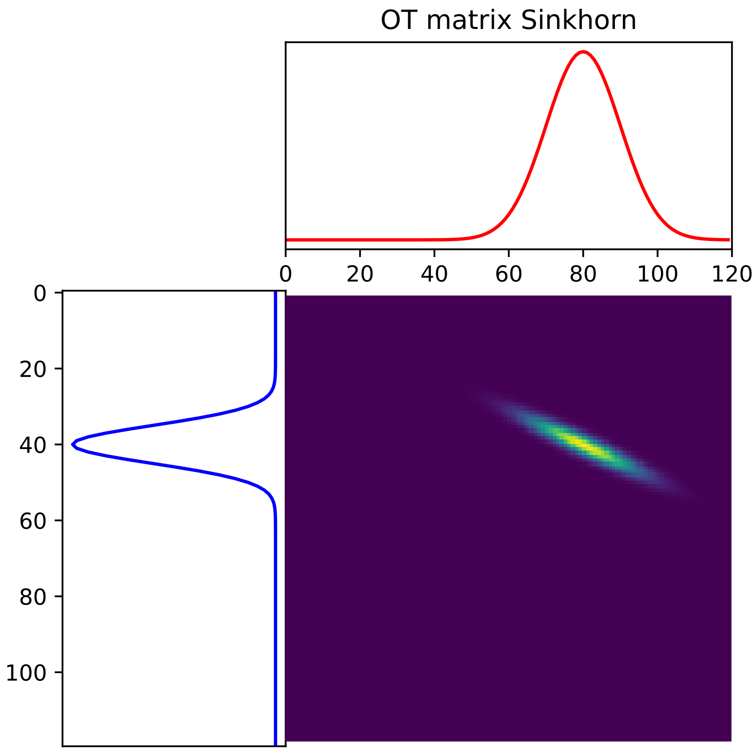


Figure 4.9 Gaussian OT using Sinkhorn Algorithm

Definition 4.2.5 (Regularized OT [61])**.** *Given a discrete measure as introduced in Equation 4.11, then we get*

$$\mathcal{L}_C^{\varepsilon}(\alpha, \beta) := L_{\mathbf{C}}^{\varepsilon}(\mathbf{a}, \mathbf{b}), \tag{4.31}$$

for the definition of optimal transport. Again, $\mathbf{C}_{i,j} = c(x_i, y_j)$ *defines the cost matrix with respect to the locations* (x_i, y_j) *supporting the input measures.*

Now, it can be shown that the solution to the regularized optimal transport problem in Def. 4.2.5 has a specific form parameterized with $n + m$ variables.

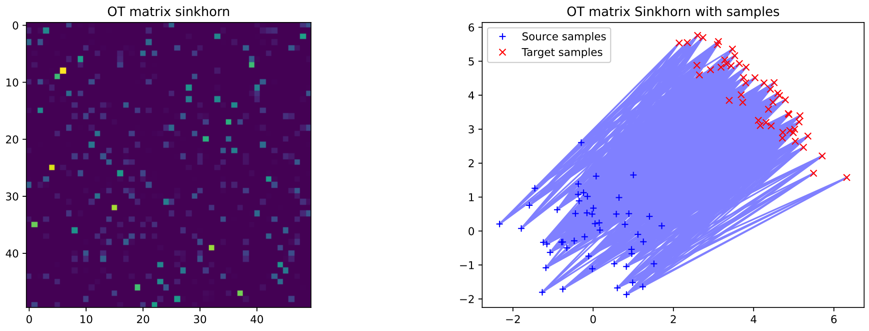
Proposition 4.2.6. *There exists a unique solution to the problem in Def. 4.2.5 [61]. The solution has the form*

$$\forall (i, j) \in [n] \times [m], \mathbf{P}_{i,j} = \mathbf{u}_i \mathbf{K}_{i,j} \mathbf{v}_j, \tag{4.32}$$

where the tuple $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$ *consists of two unknown scaling variables* $\mathbf{u}$ *and* $\mathbf{v}$ *[61].*

Equation 4.32 gives a factorization of the optimal solution for the regularized optimal transport problem. This solution can be rewritten in matrix form as $\mathbf{P} = \text{diag}(\mathbf{u})\mathbf{K}\text{diag}(\mathbf{v})$. As a result, the scaling variables $\mathbf{u}$ and $\mathbf{v}$ need to fulfill the following set of equations to satisfy the mass conservation constraints in Equation 4.15 of $\mathbf{U}(\mathbf{a}, \mathbf{b})$:

$$\text{diag}(\mathbf{u})\mathbf{K}\text{diag}(\mathbf{v})\mathbb{1}_m = \mathbf{a} \text{ and } \text{diag}(\mathbf{v})\mathbf{K}^T\text{diag}(\mathbf{u})\mathbb{1}_n = \mathbf{b}. \tag{4.33}$$



(a) OT matrix using Sinkhorn

(b) Sample distribution and OT matrix

Figure 4.10 Result 2D OT of two empirical distributions

Exploiting the fact that $\operatorname{diag}(\mathbf{v})\mathbb{1}_m = v$ the set of equations can be simplified to

$$\mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \mathbf{a} \text{ and } \mathbf{v} \odot (\mathbf{K}^T\mathbf{u}) = \mathbf{b}, \tag{4.34}$$

where $\odot$ represents the entrywise multiplication of vectors. A solution to this set of equations can be found iteratively, meaning that $\mathbf{u}$ is modified first to satisfy the left hand side of Equation 4.34 and then $\mathbf{v}$ to satisfy the right-hand side afterwards. Initialized with $\mathbf{v}^0 = \mathbb{1}_m$, both updates define the Sinkhorn algorithm:

$$\mathbf{u}^{(l+1)} := \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}}^{(l)} \text{ and } \mathbf{v}^{(l+1)} := \frac{\mathbf{b}}{\mathbf{K}^T\mathbf{u}^{(l+1)}}. \tag{4.35}$$

Given $\tau > 0$, this algorithm is a τ-approximate algorithm in $\mathcal{O}(n^2 \log(n)\tau^{-3})$ iterations, meaning that $\mathcal{O}\left(||\mathbf{C}||_\infty^3 \log(n)\tau^{-3}\right)$ Sinkhorn iterations are sufficient to guarantee $\langle \mathbf{P}, \mathbf{C} \rangle \leq L_{\mathbf{C}}(\mathbf{a}, \mathbf{b}) + \tau$. A detailed proof of this fact can be found in [2]. Figure 4.10a shows the OT matrix resulting from the Sinkhorn iterations for two sample distributions as illustrated in Figure 4.10b.

4.2.1.4 Optimal transport barycenters

To guarantee smooth pattern transitions between arbitrary communication scenarios defined by probability distributions representing ever-changing sequences of link data rates, there is a special interest to compute the mean or so called barycenter between several distributions. In this case we are assuming optimal transport distances as introduced in the last section s.t. the problem can be formulated as a convex program and as a result we can proof that an optimal solution exists.

Thus, assume that $\{\mathbf{b}_s\}_{s=1}^S$ is a histogram s.t. $b_s \in \Delta_{n_s}$ and $\lambda \in \Delta_S$ is the corresponding set of weights. A Wasserstein barycenter can be computed by solving the following optimization problem:

$$\min_{\mathbf{a} \in \Delta_n} \sum_{s=1}^{S} \lambda_S L_{\mathbf{C}_s}(\mathbf{a}, \mathbf{b}_s). \tag{4.36}$$

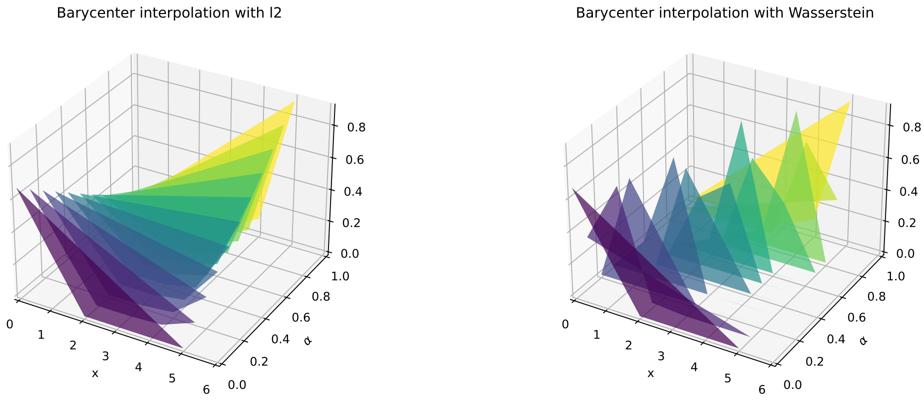


Figure 4.11 Barycentric interpolation with L^2 and Wasserstein metric

As referring to Def. 4.2.5 this problem can also be solved by entropic smoothing and approximation:

$$\min_{a \in \Delta_n} \sum_{s=1}^{S} \lambda_S L^{\varepsilon}_{\mathbf{C}_s}(\mathbf{a}, \mathbf{b}_s), \varepsilon > 0. \tag{4.37}$$

Benamou et al. [6] proposed a very efficient approach to solve this problem. The idea is to rewrite the Equation 4.37 as Bregman projection problem:

$$\min_{\mathbf{P}_{s s}} \left\{ \sum_s \lambda_s \varepsilon \mathbf{KL}(\mathbf{P}_s | \mathbf{K}_s) : \forall s, \mathbf{P}_s^T \mathbb{1}_m = \mathbf{b}_s, \mathbf{P}_1^T \mathbb{1}_1 = \mathbf{P}_S^T \mathbb{1}_S \right\}, \tag{4.38}$$

where $\mathbf{K}_s := \exp\left(\frac{-\mathbf{C}_s}{\varepsilon}\right)$. This problem can be seen as special case of the generalized Sinkhorn from 4.34. Here, the scaling form reduces to

$$\mathbf{P}_s = \operatorname{diag}(\mathbf{u}_s)\mathbf{K}\operatorname{diag}(\mathbf{v}_s), \tag{4.39}$$

with sequential updates:

$$\forall s \in [1, S], \quad \mathbf{v}_s^{(l+1)} := \frac{\mathbf{b}_s}{\mathbf{K}_s^T \mathbf{u}_s^{(l)}}, \tag{4.40}$$

$$\forall s \in [1, S], \quad \mathbf{u}_s^{(l+1)} := \frac{\mathbf{a}^{(l+1)}}{\mathbf{K}_s \mathbf{v}_s^{(l+1)}}, \tag{4.41}$$

$$\text{where } \mathbf{a}^{(l+1)} := \prod_s (\mathbf{K}_s \mathbf{v}_s^{(l+1)})^{\lambda_s} \tag{4.42}$$

This Sinkhorn iterations define the basis for the Alg. 6 defined in Sec. 4.2.1.5 and enables us to compute smooth pattern transitions efficiently [6].

```
Input: $\vec{X}_t$,$S$,$B_0$, $B_n$, $n$, $\Lambda$, reg= $0.0015$,$\varepsilon = 1e^{-9}$,$\alpha$
Output: $\mathcal{P}$, $\theta$
  Initialization :
  $\mathfrak{B} \leftarrow []$
  $\theta(0) \leftarrow 0$
  $\theta(n) \leftarrow n$
  $j \leftarrow 1$
  $\alpha_L \leftarrow \mathrm{linspace}(0, 1, \alpha)$
  Procedure :
  for $\alpha_l$ in $\alpha_L$ do
    baryWass $\leftarrow$ zeros$((n, n))$
    $i \leftarrow 0$
    for $i < n$ do
      $p \leftarrow A[i][:]$
      $q \leftarrow B[i][:]$
      $baryWass[i] \leftarrow$ wassersteinBarycenter1d$(p, q, \alpha_l, reg, \varepsilon)$
    end for
    $\mathfrak{B}[j] \leftarrow baryWass$
    $\theta(j) \leftarrow j$
    $\mathcal{P} \leftarrow \mathcal{P} \cup (S, \Lambda(j), \mathfrak{B}[j], \vec{X}_{t+j})$
  end for
  return $\mathcal{P}$, $\theta$
```

Algorithm 5 : BarycentricInterpolation

4.2.1.5 Smoothing pattern transitions

As introduced in the last section, OT is often used to measure similarity between distributions, even if they do not share the same support. Introducing Sinkhorn's algorithm and Bregman projections provides a very efficient tool to measure and optimize similarity between empirical distributions. The following application exploits these facts to measure the similarity of patterns generated by the in-homogeneous Markov $Model_B$. Moreover, the concept of Wasserstein barycentric interpolation will be used to generate a set of cross-distributions defining smooth pattern transitions between patterns of data rate changes, thus replacing the analytical functions defined in Section 4.1.

To this end, we assume that we are given the start and target distributions B_0 and B_n of two patterns P_0 and P_n. Moreover, let n define the number of cross-distributions generated by the interpolation, reg be the regularization, and ε be the epsilon parameter for stabilizing the regularized OT. Using the weight parameter α for each histogram, we can compute the 2-dimensional Wasserstein barycenters of the source and target distribution by iteratively taking the marginal distributions from the corresponding pattern matrix and computing the 1-dimensional Wasserstein barycenter for both marginal distributions as shown in Algorithm 5.

Once we updated the set of patterns $\mathcal{P}$ and also the transformation function θ, we can reuse the *Sample* algorithm (Algorithm 2) to sample a sequence of data rate changes representing the Wasserstein transition morphing P_0 to P_n as illustrated in Algorithm 6. Figure 4.11 shows an example of barycentric interpolation between $\bar{B}_0$

```
Input: $B_0$, $B_n$, $X_t$, $S$, $\Lambda$, $n$, $\delta$,$\lambda$,$\alpha$,$seed$
Output: Sequence of states $\Sigma$
  Initialization :
  $\mathcal{P}$,$\theta$ $\leftarrow$ BarycentricInterpolation($X_t$,$S$,$B_0$,$B_n$,$n$,$\Lambda$,$\alpha$)
  return  $Sample\left(\vec{X}_t, \mathcal{P}, \theta, \Lambda, T, \lambda, seed\right)$
```

Algorithm 6 : PatternTransition

and $\bar{B}_5$. Here, the left plot illustrates the process of moving the mass from state 0 to state 5 using the L^2 metric, which is related to the transformations done by Algorithm 3. In this case, we use the transformation matrix to decrease the mass at state zero, while increasing the mass at state 5 in parallel. In contrast, the right plot in Figure 4.11 shows a smoother way to transform state 0 to state 5 by slowly moving the mass across the nearest neighbours of each state (state 1 to 4). This visualizes why we decided to replace the transformation functions from Section 4.1 by barycentric interpolation, since it is more accurate to slowly move the mass across nearest neighbour distributions.

4.2.2 The multi-agent model

This section extends *Model$_B$* to a MDP, thus allowing choice (action space) motivated by rewards (reward function). As a result, the MDP adds one more layer of abstraction enabling partial control by an agent trying to find an optimal policy by solving a set of mathematical equations (Bellmann equation [5]). The benefit of a MDP modeling the network conditions is that the outcomes of decision making are partly random and partly controlled by the agent supporting the high uncertainty in TNs caused by the challenging environment of military operations. This extension of *Model$_B$* is motivated by the idea to automate experiments using a multi-agent system as introduced later in Section 4.2.2.

4.2.2.1 Theoretical foundations of MDPs

MDPs are discrete-time stochastic processes mostly used to model agent based decision making in scenarios, where outcomes are partly random and partly controlled by an agent [63]. More precise, similar to MCs the process is distributed over time $\mathfrak{T}$, meaning that, at each time step $t \in \mathfrak{T}$, the process is in some state $x_t \in \mathcal{X}$. In addition, an agent can take partial control of the next state X_{t+1} by choosing an action $a_t \in \mathcal{A}$, available at state x_t. Once the action is chosen, the process responses moving into a new state x_{t+1} and returns the reward $r(x_t, a_t)$ to the agent. Here, the state transition from x_t into x_{t+1} depends on the probability chosen from the distribution of an stochastic kernel κ. This stochastic kernel provides a distribution over the state space for each feasible state-action pair (x_t, a_t), denoted by G. In comparison to a MC, the next state x_{t+1} only depends on the current state x_t and the agents action a_t, but is conditionally independent of all previous states and actions and as a result the state transitions of a MDP satisfy the Markov property.

Definition 4.2.7 (Markov decision process)**.** *Define a Markov decision process is to be the six-tuple* $(X, A, \Gamma, r, \beta, \kappa)$*, where* *(i)* $\mathcal{X}$ *is the nonempty finite state space;*

(ii) $\mathcal{A}$ *is the nonempty finite action space;*

(iii) $\Gamma : \mathcal{X} \to \mathcal{A}$ *is the feasible correspondence;*

(iv) $r : G \to \mathbb{R}$ *is the reward function, with*

$$G := \{(x_t, a_t) \in \mathcal{X} \times \mathcal{A} : a \in \Gamma(x_t)\}$$

(v) $\beta \in (0, 1)$ *is the discount factor and*

(vi) $\kappa : G \to \mathcal{X}$ *defines the stochastic kernel.*

In the following we define the stochastic kernel κ from G to $\mathcal{X}$ being a family of distributions $\kappa(g, \cdot)$ over the state space $\mathcal{X}$, one for each $g \in G$. As a result, we can sample the next state from this distribution. Especially for discrete environments, like ever-changing network conditions distributed over time $\mathfrak{T}$, the main interest is in optimal strategies of one or multiple agents. One common approach to define an optimal strategy is to go over actions by policies, meaning that we are mapping states to actions. Using this strategy, we define the set of feasible policies as

$$\Pi := \left\{\pi \in A^{\mathcal{X}} : \pi(x) \in \Gamma(x) \text{ for all } x \in \mathcal{X}\right\}. \tag{4.43}$$

In other words, selecting a particular policy $\pi \in \Pi$ can be interpreted as the response of the process to state x_t given action $a_t := \pi(x_t)$ at any point in time $t\mathfrak{T}$. Moreover, if the agent commits to a policy π for the lifespan of some problem P, then the state x_t evolves by sampling x_{t+1} from $\kappa(X_t, \pi(X_t), \cdot)$. Moreover, given an initial condition $x_0 = x$, this process is an (x, κ_π)-MC for κ_π defined by

$$\kappa_\pi := \kappa(x, \pi, y) \quad (x, y, \in \mathcal{X}). \tag{4.44}$$

As a result, fixing a policy *"closes the loop"* in the state transition process and sets a given MC for policy π. Now we can define the rewards under policy π such that

$$r_\pi(x_t) := r(x_t, \pi(x_T)) = r(x_t, a_t) \tag{4.45}$$

and with this notation, it follows that the expected time reward $\mathbb{E}\left[r(x_t, a_t) | x_0 = x\right]$ can be calculated as

$$\mathbb{E}\left[r(x_t, a_t) | x_0 = x\right] = \mathbb{E}\left[r_\pi(x_t) | x_0 = x\right] = \kappa_\pi^t r_\pi(x). \tag{4.46}$$

Moreover, the lifetime value of following policy π starting from $x_0 = x$ is given by

$$
\begin{aligned}
v_\pi &= \mathbb{E}\left[\sum_{t\geq 0}\beta^t r(x_t,\pi(x_t))|x_0 = x\right] \\
&= \sum_{t\geq 0}\mathbb{E}\left[\beta^t r(x_t,\pi(x_t))|x_0 = x\right] \\
&= \sum_{t\geq 0}\beta^t\left(\kappa_\pi r_\pi\right)(x).
\end{aligned}
\tag{4.47}
$$

This function is often called *π-value function.* Now, we can use this function to define the *value function* of the MDP to be

$$v^*(x) = \sup_{\pi\in\Pi} v_\pi(x) \quad (x \in X). \tag{4.48}$$

The *value function* represents the lifetime value extracted from each state, given that the agent's behaviour is optimal at each point in time. Keeping this in mind, a policy $\pi \in \Pi$ is called optimal strategy if $v_\pi = v^*$. In other words, a policy is optimal if its lifetime value is maximal at each state in time. Now, Bellman [5] showed that that at least one optimal policy exists.

Proposition 4.2.8. *The value function v^* satisfies the Bellmann equation*

$$v^*(x) = \max_{a\in\Gamma(x)}\left\{r(x,a) + \beta\sum_{y\in\mathcal{X}} v^*(y)\kappa(x,a,y)\right\} \tag{4.49}$$

at every $x \in \mathcal{X}$. Moreover, a feasible policy π is optimal if and only

$$\pi(x) \in \arg\max_{a\in\Gamma(x)}\left\{r(x,a) + \beta\sum_{y\in\mathcal{X}} v^*(y)\kappa(x,a,y)\right\} \tag{4.50}$$

At least one such policy exists.

Proof. A detailed proof of the proposition can be found in [5]. □

This is the main principle of the theory of dynamic programming. The optimal value function v^* is a unique solution to the *Bellmann equation* and π^* is an optimal policy function if and only if it is *v^*-greedy.* Solving these Discrete Dynamic Programming (DDP) problems can be done by implementing one of the three solutions namely

- value function iteration [4],
- policy function iteration [28] or
- modified policy function iteration [59, 64].

While in value iteration only the function v^* is used and the the value π is only computed if it is needed for the calculation of v^*, policy function iteration and also modified policy function iteration combine both steps. In policy iteration, step one is performed only once for each possible state followed by a repetition of step two until convergence. This can also be interpreted as solving the linear equations by an iterative method. In addition, modified policy iteration is a variation of the policy iteration method also performing step one only once, but using an approximate solution for step two, thus resulting in less computation steps.

4.2.2.2 Theoretical foundations of statistical testing

In addition to Section 4.2.1 this Section revisits the basic foundations of statistical testing for the Kolmogorov-Smirnov Test (K-S test) and the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality, used later in this thesis to verify if the distribution of the TtR converges with respect to the duration of random link disconnections in tactical networks.

The K-S test is a non-parametric and distribution free test to compare a distribution against a reference distribution (one-sided) or to test if the Cumulative Distribution Function (CDF) of two samples are equal (two-sided). More precise, given two samples X_1 and X_2 of size n and m, the K-S test evaluates if the null hypothesis H_0 := "The samples do indeed come from the same distribution F_{exp}" is accepted or rejected by computing a distance between the Empirical Distribution Functions (EDFs) $F_{1,n}$ and $F_{2,m}$ of both samples. One major advantage of the K-S test is, compared to most of other statistical tests, that the K-S test is not restricted to normal distributed random variables and that the size of both samples does not have to be equal. This makes the K-S test applicable, even if there is no guess for the underlying distribution of both samples as it is the case for the distribution of the TtR, considered to be the target quantity in this part of the thesis. To perform the K-S test we compute the following Kolmogorov-Smirnov statistic

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{4.51}$$

and compare the value of $D_{n,m}$ against the threshold D_α given as

$$D(\alpha) := c(\alpha)\sqrt{\frac{n+m}{n \cdot m}}, \tag{4.52}$$

where the value of $c(\alpha)$ is computed by

$$c(\alpha) = \sqrt{-\ln\left(\frac{\alpha}{2}\right) \cdot \frac{1}{2}}. \tag{4.53}$$

Now, we can use the threshold D_α (Equation 4.52) to evaluate if there is strong evidence to reject the null hypothesis H_0, meaning that $D_{n,m} > D_\alpha$. We will use this test in Section 4.2.2.5 as one criteria to determine if the distribution of the TtR converges in distribution w.r.t to the disruption time.

Revisiting the K-S test, we got a procedure to test whether two random variables are following the same underlying probability distribution. Now, the DKW inequality [17] can be interpreted as an inversion of this process computing the the confidence bands of the CDF $F_n(x)$ itself. Assuming that $X_1, \ldots, X_n$ are real-valued independent and identically distributed (i.i.d.) random variables with CDF $F(\cdot)$ and EDF

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{X_i \leq x}, \quad x \in \mathbb{R} \tag{4.54}$$

respectively, the DKW equation gives an upper bound for the probability that the difference of F_n and F is larger than a given constant $\varepsilon > 0$. More formally the DKW states that

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}}(F_n(x) - F(x)) > \varepsilon\right) \leq \exp -2n\varepsilon^2 \quad \forall\varepsilon \geq \sqrt{\frac{1}{2n}\ln 2} \tag{4.55}$$

for the one-sided estimate and

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}}|F_n(x) - F(x)| > \varepsilon\right) \leq \exp -2n\varepsilon^2 \quad \forall\varepsilon \geq 0 \tag{4.56}$$

for the two-sided estimate, which can be used to generate CDF-based confidence bands to analyze similarity of a set of different EDFs. To this end, we can define the interval containing the original CDF with probability $1 - \alpha$ by

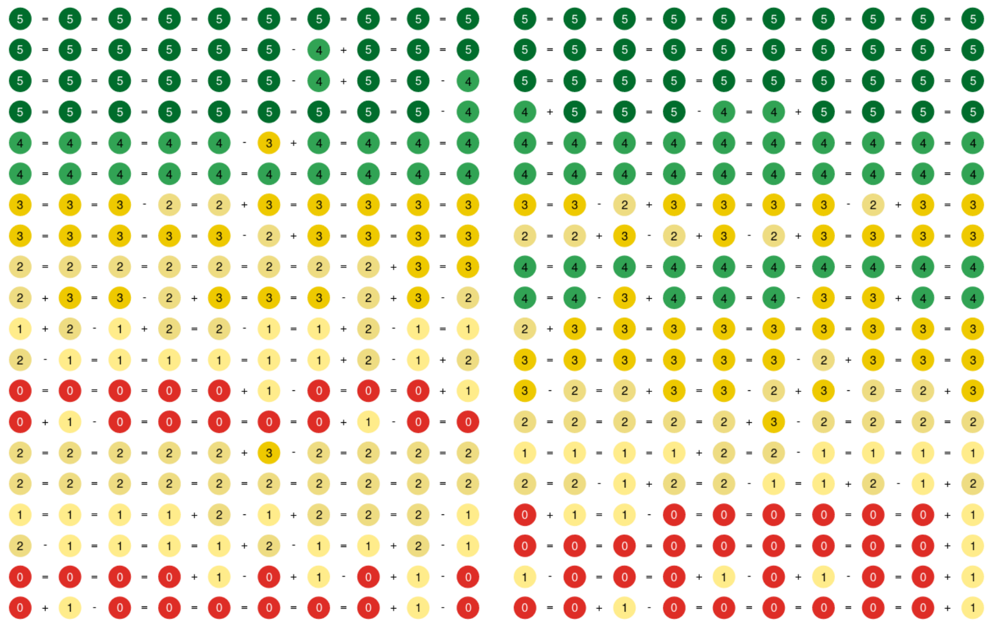
$$F_n(x) - \varepsilon \leq F(x) \leq F_n(x) + \varepsilon, \text{ where } \varepsilon = \sqrt{\frac{\ln\frac{2}{\alpha}}{2n}} \tag{4.57}$$

In the following, we will apply this foundations combined with the Wasserstein and Energy distance to define a multi-agent system verifying that the the null hypothesis H_0 can be accepted for comparing the CDF of the TtR after unplanned link disconnections.

4.2.2.3 The MDP for ***Problem*** B

Now, we apply the theoretical foundations introduced in Section 4.2.2.1 to define a MDP := $(\mathcal{S}, \mathcal{A}, \Gamma, r, \beta, \kappa)$ solved by an Agent A_{MDP} creating ever-changing network conditions. Recall that the original motivation to extend *Model*$_B$ to a MDP comes from the fact that the outcomes of decision making are partly random and partly controlled by the agent, thus supporting the high uncertainty in TNs caused by the challenging environment of military operations. Moreover, the reward function enables us to incorporate system feedback in the modeling process, which is exploited to define a multi-agent system learning the distribution of the TtR after unplanned link disconnections. To this end, let $\mathcal{S} = \{s_1, \ldots, s_N\}$ be the set of states representing the probability distributions of N finite MCs and $\mathcal{A} = \{a_1, \ldots, a_N\}$ be the set of actions, where action a_n indicates that the agent recommends to move to distribution s_n in the following state. One exemplary configuration could be to define the state space $\mathcal{S}$ to be the set of distributions representing the stable conditions defined by $\bar{B}_0, \bar{B}_5$ and δ cross-distributions $B_1, \ldots, B_\delta$ generated by Wasserstein barycentric interpolation [6] as introduced in Section 4.2.1.4.

The state transitions of the MDP are sampled from a stochastic kernel κ from G to $\mathcal{S}$ defined by a family of distributions $\kappa(g, \cdot)$ over state space $\mathcal{S}$, one for each $g \in G$. In other words, the stochastic kernel introduces uncertainty into the state transition process, meaning that given a state-action pair (a_t, s_t), the next state s_{t+1} is determined by sampling from the respective distribution $g_t = (s_t, a_t) \times a_t \in G$. In this thesis, there are no restrictions on the set of feasible actions $a_t \in A$



(a) Loop L_3 ~3.19 kbps ±3.09

(b) L_4 ~3.87 kbps ±3.80

Figure 4.12 Two loops L_3 and L_4 created by the MDP

, meaning that independent of the current state $s_t \in \mathcal{S}$, any action $a_t \in \mathcal{A}$ is a feasible correspondence. Finally, the reward function $r : G \rightarrow \mathbb{R}$ is assumed to be a real valued and defines the intuition behind the optimal policy. Now, agent A_{MDP} computes an optimal policy v^* for the MDP by DDP using modified policy iteration. Remembering that, the optimal policy v^* fixes on action for each state of the MDP and thus defines a finite MC, it can be interpreted as an instance of the in-homogeneous Markov *Model*$_B$. As a result, one can use the *Sample Alg.* from Section 4.1 to sample a sequence the data rate changes of a communication scenario distributed by the optimal policy.

Figure 4.12 shows two exemplary loops L_3 and L_4 generated by the MDP. Both loops represent a metamorphosis from $\hat{B}_0$ to $\hat{B}_5$ using $\delta = 4$ cross-distributions and are computed throwing 10 samples from the optimum policy $v^* = [\hat{B}_0, B_1, B_2, B_3, B_4, \hat{B}_5]$ with respect to the stochastic kernel κ and sampling 20 network states from each of the resulting probability distributions afterwards. For example, L_3 represents a sequence of network states by sampling 20 states from the series of probability distributions given by $v_3 = [\hat{B}_0, B_1, B_2, B_0, B_1, B_2, B_3, B_4, \hat{B}_5, \hat{B}_5]$ sampled from the optimal policy v^*. This illustrates how the stochastic kernel can be used to model uncertainty by allowing random transitions after a state-action pair is chosen. We can observe this phenomena for L_3 after state 60, where the probability distribution, defining the states changes, jumps from B_2 back to B_0 again and the loop is starting from the beginning B_0 again. The same holds for L_4 and $v_4 = [\hat{B}_0, \hat{B}_0, B_1, B_2, B_3, B_4, B_3, B_4, \hat{B}_5, \hat{B}_5]$ after state 120, where the distribution jumps from B_3 back to B_4 and then back to B_3 again.

4.2.2.4 The multi-agent model

Using the MDP from the previous Section, we are now able to define the multi-agent *Model$_{B^*}$* composed of two agents collaborating to quantify the distribution for the TtR after link disconnections, as shown in Figure 4.13. More precise, agent A_{MDP} creates the changes in the link data rate including link disconnections, while in parallel the second agent A_{H_0} verifies if the null hypothesis H_0 := *"The CDFs of the TtR are indeed equal"* is accepted or rejected by analyzing the metrics of a configuration of different statistical tests defined by $\mathcal{D}_{H_0}$ (Section 4.2.2.5). As referring to Figure 4.13, these metrics are analyzed at two different stages: First, the hypothesis is tested for fixed disruption time, meaning that the agent tests if the TtR for a fixed disruption time remains the same over time (locally). Second, the test is executed between distributions of different disruption time, testing if the distribution of the TtR converges independently of the disruption time (globally). Moreover, the tests can be applied to different IP data flows, namely broadcast/unicast and overlay, which offers a wide range of configuration options.

To this end, we initialize agent A_{MDP} with a MDP $:= (\mathcal{S}, \mathcal{A}, \Gamma, r, \beta, \kappa)$ solving *Problem B* as referring to Section 4.2.2.3 and agent A_{H_0} by defining the 9-tuple $(\mathcal{S}_{H_0}.\mathcal{A}_{H_0}, \mathcal{D}_{H_0}, \Lambda, \vec{X}_0, \varepsilon, m, d, d_{\text{inc}})$. In this formulation, the state space $\mathcal{S}_{H_0}. := \{s_0, s_1, s_2\}$ of the agent consists of three states, where state s_0 means the *"H_0 rejected locally"* yet, state s_1 the *"H_0 accepted locally"*, and state s_2 the *"H_0 accepted globally"*. Moreover, the action space $\mathcal{A}_{H_0} = \{a_1, a_2, a_3\}$ is given by the three actions a_1 := *"Rerun experiment with same configuration"*, a_2 := *"Increase disruption time and run a new experiment"* and a_3 := *"Stop procedure and output the final distributions for the TtR"*. Using this state and action space, the multi-agent system operates as follows.

Agent A_{H_0} starts in state s_0 (*"H_0 rejected locally"*) with initial disconnection time d checking if the last m experiments with different disconnection time are passing the statistical tests from Section 4.2.2.5, meaning that the null hypothesis can be accepted globally. But, since we did not execute any experiments before, this is not the case. As a result, A_{H_0} asks A_{MDP} to generate a new experiment using the optimal policy v^* computed from the MDP. It should be noted that the statistical tests as well as the distribution for the TtR are automatically computed by our monitoring service and that the results of the tests are available to agent A_{H_0} after each experiment.

Once a new experiment is executed, the A_{H_0} determines if the null hypothesis H_0 can be accepted locally, meaning that the distribution of the TtR before and after executing the experiment is similar within a 95% confidence interval, which is evaluated by using the statistical tests from $\mathcal{D}_{H_0}$. If not, agent A_{MDP} is triggered again. This procedure is repeated until the null hypothesis can be accepted locally. If this is the case, agent A_{H_0} stores the final CDF and the disconnection time in $\mathcal{F}$, switches to state s_1 and checks if the null hypothesis can also be accepted globally. If not, A_{H_0} increases the disconnection time $d + d_{inc}$ and cleans his cache holding all the information from the last series of experiments, except from the information stored in $\mathcal{F}$.

Since the agent recognizes that, given the new disconnection time, H_0 is not accepted yet, it will switch back to state s_0, thus executing a new series of experiments until

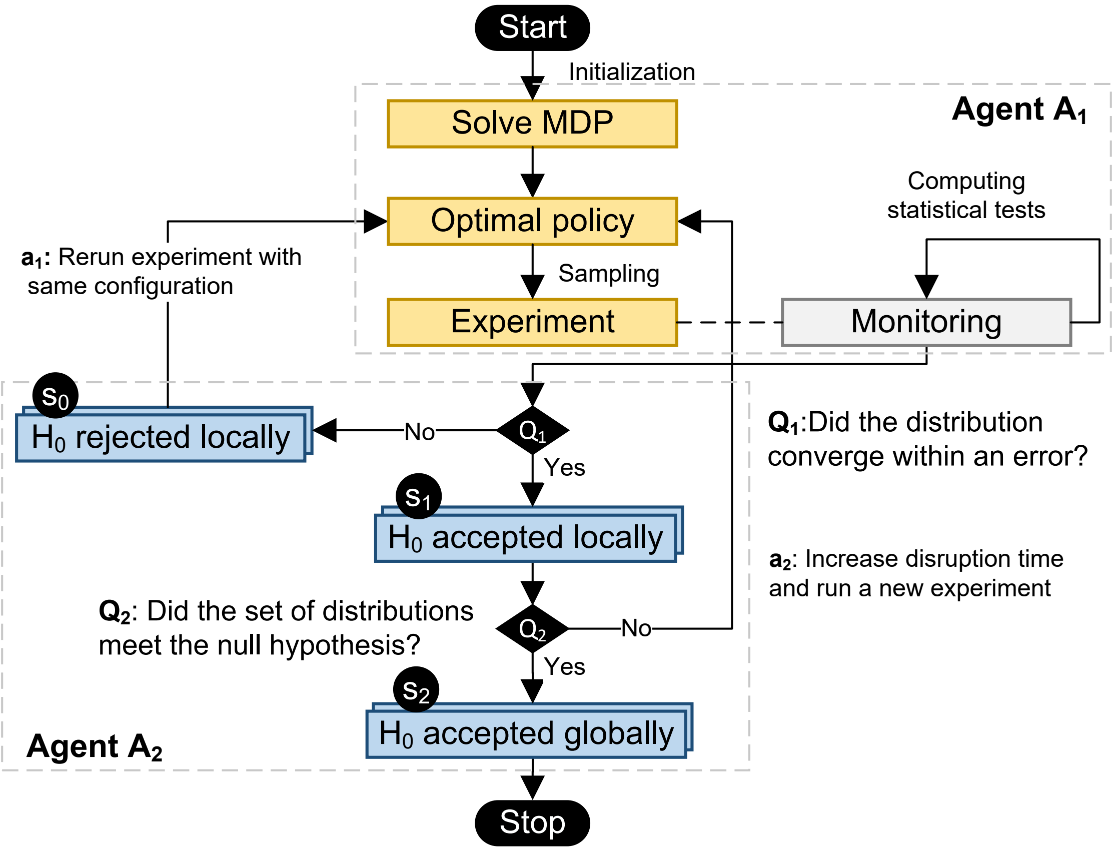

Start
Initialization
Agent A₁
Solve MDP
Computing statistical tests
Optimal policy
a₁: Rerun experiment with same configuration
Sampling
Experiment
Monitoring
S₀
H₀ rejected locally
No
Q₁
Q₁:Did the distribution converge within an error?
Yes
S₁
H₀ accepted locally
a₂: Increase disruption time and run a new experiment
Q₂: Did the set of distributions meet the null hypothesis?
Q₂
No
Yes
S₂
H₀ accepted globally
Agent A₂
Stop

Figure 4.13 Two-Agent Model to compute the distribution of the TtR [44].

it reaches state s_1 again. This process is repeated until the agent reached state s_1 and m consecutive distributions in $\mathcal{F}$ are passing the statistical tests, meaning that the null hypothesis for the distribution of the TtR is accepted, independent of the disconnection time.

4.2.2.5 Testing the null hypothesis H_0

To test the null hypothesis locally and globally the second agent A_{H_0} performs statistical tests that are equivalent to showing that the EDF of the TtR is following a respective mode of convergence, which is called *convergence in distribution or weak convergence*. To define this mode of convergence, let $X_1, \dots, X_n, \dots, X_N$ be a series of random variables describing the time to resume t_{resume} after N link disconnections. This means that the state space $\Omega \subset \mathbb{N}$ of the random variables is represented by the number of link disconnections in a series of experiments, meaning that the random variables $X_i : \Omega \to E := \mathbb{R}, i \in \{1, \dots, N\}$ are real-valued. Using the corresponding CDF $F_1, \dots, F_n, \dots, F_N$ of the random variables we can define the mode of convergence as follows.

Definition 4.2.9 (Convergence in distribution)**.** *Let $X_1, \dots, X_N$ be a series of real-valued random variables, then we say the series is converging in distribution or weakly to some random variable X if the following requirement is fulfilled*

$$\lim_{n \to \infty} F_n(x) = F(x) \tag{4.58}$$

for every number $x \in \mathbb{R}$ at which F is continuous. Here, F_n and F are the CDFs of the random variables X_n and X, respectively.

Section 5 shows exemplary thresholds for the p-value of the K-S test, Wasserstein and Energy tests representing this mode of convergence.

4.3 *Model* A|B: The stochastic uncertainty model

In this section we propose a solution to *Problem A*$|B$ by introducing a stochastic uncertainty model that can measure and finally improve the robustness of TS by calculating the probability of packet/message delivery (p_1/p_2) in TN over ever-changing communication scenarios. To this end, we state and solve an optimization problem maximizing the probability of packet/message delivery by computing the minimum amount of redundancy that has to be added to the user data flow, s.t. the probability for message delivery converges to an upper bound. The proposed solution follows the investigation [39], which was developed and published as part of this thesis.

4.3.1 Computing the probability of packet and message delivery

4.3.1.1 Computing the ground truth p_0

We start, sketching a solution for computing the probability of single packet delivery $p_0(T_w)$ in a discrete and finite time window T_w (sec). This is the first and fundamental step to compute the probabilities p_1 and p_2 later on, since we will reuse this probability in the respective formulas. To do so, we assume that the TS is equipped with a monitoring service and that we have access to all the the features collected during the communication scenarios generated by $Model_B$ and/or $Model_{B^*}$. Moreover, we assume that the network conditions $\mathcal{N}$ of the system are defined by a communication scenario $\mathcal{C}$ as defined in 4.1.4 and generated by $Model_B$ or $Model_{B^*}$. Now, we can define the time window $T_w(t_{\text{start}}, t_{\text{end}})$ starting at time t_{start} and ending at time t_{end} ($t_{\text{start}} < t_{\text{end}} \in \mathfrak{T}$) by using the time distribution function λ for updating the link data rates

$$T_w(t_{\text{start}}, t_{\text{end}}) := \left[\max\left(0, \sum_{i=1}^{t_{\text{start}}} \lambda(i) + 1\right), \sum_{i=1}^{t_{\text{end}}} \lambda(i)\right]. \tag{4.59}$$

Assuming that the probability of delivering a packet $p_0(T_w)$ is proportional to the amount of bits received within the time window $T_w(t)$ and the packet size is distributed by κ, the *optimal* data rate $b_{\text{opt}}(t)$ for almost sure delivery can be calculated using the ratio

$$b_{\text{opt}}(T_w(t_{\text{start}}, t_{\text{end}})) = \frac{\kappa \text{ (kb)}}{|T_w(t_{\text{start}}, t_{\text{end}})| \text{ (s)}}. \tag{4.60}$$

Now, we can compute the *ratio* between the data rate $b(t)$ at time t and the *optimal* data rate $b_{\text{opt}}(t)$ for a given time window T_w (Equation 4.61).

$$b_{\text{ratio}}(t_{\text{start}}, t_{\text{end}}, T_w) = \frac{\sum_{t=t_{\text{start}}}^{t_{\text{end}}-1} \frac{T_w(t,t+1)}{|T_w(t_{\text{start}},t_{\text{end}})|} b(t)(\text{kbps})}{b_{\text{opt}}(T_w(t_{\text{start}}, t_{\text{end}}))(\text{kbps})} \tag{4.61}$$

In the following we will use the short cuts T_w for $T_w(t_{\text{start}}, t_{\text{end}})$, $T_w(t)$ for $T_w(t, t+1)$ $b_{\text{opt}}(T_w)$ for $b_{\text{opt}}(T_w(t_{\text{start}}, t_{\text{end}}))$ and b_{ratio} for $b_{\text{ratio}}(t_{\text{start}}, t_{\text{end}}, T_w)$. As a result from (4.60) and (4.61), we can use b_{ratio} to introduce a function $g(t, b(t), b_{\text{opt}}(T_w), t_{\text{start}}, t_{\text{end}})$ that computes an initial guess for the probability of packet delivery at *layer 0*

$$g(t, b(t), b_{\text{opt}}(T_w), t_{\text{start}}, t_{\text{end}}) = \begin{cases} 0, & \textit{if}\; b(t) = 0 \quad \forall t \in [t_{\text{start}}, t_{\text{end}}] \\ b_{\text{ratio}}, & \textit{if}\; \sum_{t=t_{\text{start}}}^{t_{\text{end}}-1} \frac{T_w(t)}{|T_w|} b(t) < b_{\text{opt}}(T_w) \\ 1, & \textit{if}\; \sum_{t=t_{\text{start}}}^{t_{\text{end}}-1} \frac{T_w(t)}{|T_w|} b(t) \geq b_{\text{opt}}(T_w). \end{cases} \tag{4.62}$$

This function can be interpreted as follows. If $g = 0$, this means that the data rate $b(t)$ is zero for the entire time window T_w and thus no packets can be transmitted. As a result, the probability of single packet delivery remains zero in this case. Assuming $g = 1$, then the capacity of the link for given time window T_w can be greater or equals the size κ of the packet and as a result $p_0(t) > 0$. Moreover, if $g = b_{\text{ratio}}$, then we can solve the equation $|T_w(t)| \cdot b/\kappa = 1 - g(t, b(t), b_{\text{opt}}(T_w), t_{\text{start}}, t_{\text{end}})$ either for parameter $|T_w|$ or parameter b to adjust the system parameters fulfilling $g = 1$ resulting in $p_0(t) > 0$. Now, let us consider the most interesting case $g = 1$ and $b(t) > b_{\text{opt}}(T_w(t))$ or more precisely $b(t) = (1+\epsilon) \cdot b_{\text{opt}}(T_w(t))$. In this case, the link is able to handle an additional amount of data $\epsilon \cdot b_{\text{opt}}(T_w(t))$, which can be used to increase the probability of packet delivery $p_0(t)$ by adding an amount of redundancy $r(t) = \epsilon \cdot b_{\text{opt}}(T_w(t))$ with $\epsilon = (b(t) - b_{\text{opt}}(T_w(t)))/b_{\text{opt}}(T_w(t))$.

As a result from the analysis of function g in the last paragraph, we conclude that $p_0(t)$ can be written as a conditional probability depending on the data rate $b(t)$, the packet size κ and the time window $T_w(t)$

$$p_0(t) = \mathbb{P}(X = 1 | b(t), T_w(t), \kappa). \tag{4.63}$$

To guarantee $p_0(t) > 0$ even under ever-changing network conditions in a laboratory with military radios, we exploited the properties of function g by varying the parameters of g and as a result also of p_0 during a set of experiments over stable network conditions. Figure 4.14 shows exemplary values of $p_0(T_w(t))$ measured over stable network conditions in our laboratory environment at Fraunhofer FKIE. This baseline results are used to compute the probability of message delivery p_2 even under ever-changing network conditions in the next section. The figure shows to plots varying the rate at the sender radio, meaning that we sent 0.05 (left) and 2 (right) packets per second. To this end, we used a packet size of 1.3 kB ($\kappa = 1.3$ kB) and considered a time window $T_w(t)$ from 0 to 300 seconds.

4.3.1.2 Computing the probabilities p_1 and p_2

Now, that we have the baseline results for the probability of single packet delivery p_0, we are prepared to compute both, the probability of packet delivery p_1 for more than one packet and the probability of message delivery p_2. For this purpose, we

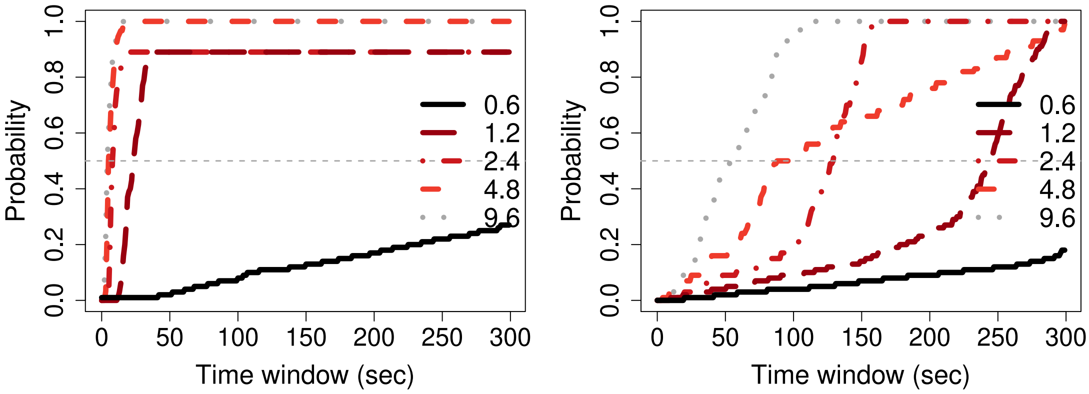


Figure 4.14 Probability p_0 for five nominal data rates using g [39].

select a technique called *Reed Solomon Code* ([25, 80]), which assumes that the message consists of k data and $n-k$ redundant packets. Using this technique, a message can be successfully delivered if at least k packets are delivered, since the lost packets can be re-transmitted by sending a selective acknowledgement, which indicates the sequence numbers of the lost packets. Moreover, if more than k packets are successfully delivered no re-transmission is required. Since each single re-transmission comes with an additional overhead and the network consisting of long range links using VHF or HF communication technologies is already fighting with low data rates and high latency, the goal is to avoid this additional overhead by adjusting the system parameters to maximize the probability that at least k packets are transmitted. Exploiting the fact that we already measured p_0 for stable conditions and successfully sending and receiving k out of n depends on the probability p_0, if errors in the network are assumed to happen independently, p_1 can be computed by a binomial distribution as shown in Equation 4.64.

$$p_1 = \mathbb{P}(X = k) = \binom{n}{k} \cdot p_0^k \cdot (1 - p_0)^{n-k} \qquad ,0 \leq k \leq n \tag{4.64}$$

Here, p_0 is the probability that a single packet will be delivered and X is a random variable measuring the number of successfully transmitted packets. The term p_1 is due to the layer, where the probability is calculated. To be specific, it means that p_1 indicates that we compute the probability that a packet is successfully transmitted in the IP-Layer (layer 1). The equation holds, because we have $\binom{n}{k}$ possibilities to sample a set of k packets out of $\Omega = \{1, \ldots, n\}$ packets.

Now, assume that the network conditions $\mathcal{N} \in \mathcal{C}$ of a communication scenario at the tactical edge are defined as the composition of patterns of data rate changes $P_1, \ldots, P_M \in \mathcal{P}$. Remember that a pattern of data rate change is the 4-tuple $P = (S, \Lambda, B_P, \vec{X}_0)$, where S is the state space representing the different data rates supported by the tactical radios, Λ is the pattern length, B_P is a stochastic matrix representing the distribution of the state transitions of a MC and $\vec{X}_0$ is the $|S|$-dimensional initial state vector. In this case, the probability that the random variable X is in state k is highly dependent on the current link state (link data rate) and thus dependent on the patterns defined in $\mathcal{P}$. As a result, if E_1 is the event $X \geq k$, meaning that the message can be successfully delivered in the first round and E_2 event that the data rate changes $s(t)$ at time t follow the distribution of pattern P_t, we get

$$\mathbb{P}(E_1|E_2) = \mathbb{P}(X \geq k|s(t) = P_t) = \frac{\mathbb{P}(X \geq k \cap s(t) = P_t)}{\mathbb{P}(s(t) = P_t)} \tag{4.65}$$

This approach can be extended to compute the probability p_2 of message delivery (at *layer 2* in Figure 3.1) during a communication scenario $\mathcal{C}$ by finding an optimum configuration for

$$p_2 = \mathbb{P}(X \geq k|\mathcal{C}) = \sum_{t=1}^{T} \mathbb{P}(X \geq k_t|s(t) = P_t) \tag{4.66}$$

in terms of k_t and fulfilling the constraint $\sum_{t=1}^{T} k_t = k$ at the same time.

4.3.2 Maximizing the probability of message delivery

Finally, we can use the probability $p_0(T_w)$ and the properties of g to maximize the probability of message delivery in ever-changing communication scenarios by adding redundancy. For this purpose, we assume that we are given a communication scenario $\mathcal{C}$ consisting of the network conditions $\mathcal{N}$ and a uniform user data-flow $\mathcal{U}$ composed by messages consisting of k data packets of packet size κ. Furthermore, we consider that we can access the probabilities $p_0(T_w(t)) = \mathbb{P}(X = 1|b(t), T_w(t), \kappa)$ for different time windows $T_w(t)$ with maximum size $t_{w_{max}}$ as shown in Figure 4.14. With this requirements, we can state an optimization problem to find the maximum value of p_2 given $\mathcal{C} = (\mathcal{N}, \mathcal{U})$:

Problem 4.3.1 (Maximizing the probability of message delivery with a minimum amount of redundancy [39]). *Let $F_{X|\mathcal{C}}(n = r + k; p_0; k)$ be the objective function describing the binomial distribution, in (4.66), of successful message delivery given communication scenario $\mathcal{C}$ and the number of data packets k, find the optimum amount of redundancy r s.t.*

$$\begin{aligned}
&\max_{r} F_{X|\mathcal{C}}(n = r + k; p_0; k)\\
=&\min_{r} 1 - F_{X|\mathcal{C}}(n = r + k; p_0; k)\\
=&\min_{r} 1 - \sum_{t=1}^{T} f_{X|P_t}(n_t = r_t + k_t; p_0; k_t)\\
&\textit{subject to:}\\
&g_1 : \sum_{t=1}^{T} k_t = k, \; g_2 : \sum_{t=1}^{T} r_t = r,\\
&g_3 : n_t - k_t \geq 0, \; g_4 : r_t, n_t, k_t \geq 0.
\end{aligned} \tag{4.67}$$

In this formulation, the patterns $P_t \in \mathcal{P}$ of link data rates are generated by the matrices $B_P \in P_t$ defining the network conditions $\mathcal{N}$ of communication scenario $\mathcal{C}$.

Moreover, the parameters k_t defines the number of packets that have to be transmitted during the communication scenario. The idea is to find an optimal amount of redundancy r to send at least $X = k$ data packets during a scenario composed of different patterns P_t meaning that r maximizes $\mathbb{P}(X \geq k_t|P_t)$ by fulfilling the constraint $X = \sum_{t=1}^{m} k_t \geq k$.

Algorithm 7 shows a solution for this problem (4.67) by sampling the sequence of data rate changes Σ and the patterns P_t describing the network conditions of the communication scenario C. In the following steps, the algorithm computes the probability of transmitting k_t data packets for each possible combination of time windows $t_w \leq t_{w_{max}}$, pattern P_t and number of data packets $k_t \leq k$. Afterwards, the optimum amount of redundancy r_t that can be added to the system is computed as referring to Section 4.3.1.1 and the corresponding probabilities are stored in the data structure S_k and added to S_{pt} afterwards. In sequence, the algorithm computes all possible configurations maximizing $\sum_{t=1}^{T}(\mathbb{P}(X \geq k_t|P_t))$, such that $\sum_{t=1}^{T} k_t \geq k$ and each pattern P_t is considered only once. Finally, the algorithm keeps the optimal configuration, which is the one with a minimum amount of redundancy $r_{min} = \sum_{t=1}^{T} r_t = n_t - k_t$. The following theorem proofs that for any communication scenario $\mathcal{C}$ generated by $Model_B$, there exists an algorithm that finds a solution for the optimization problem stated in (4.67).

Theorem 4.3.2 (Optimized message delivery [39]). *Given the non-empty, ever-changing network condition $\mathcal{N} = (\vec{X}_0, S, \mathcal{P}, \theta, \Lambda, \mathfrak{T}, \Sigma)$ of a communication scenario $\mathcal{C}$, the probabilities $\mathbf{P}_0$ for different end-to-end delays $1, \ldots, t_{w_{max}}$ over stable system conditions, the number $k > 0$ of data packets defining the length of a message Msg, the distribution of the the packet size κ and the maximum end-to-end delay $t_{w_{max}}$, MAXPROB (Alg. 7) computes an optimum solution to the problem in (4.67).*

Proof. For the proof of the theorem we refer to investigation [39], which was published as part of this thesis.

□

4.3.3 Multi-layer stochastic uncertainty model

Assuming that we are given the probabilities $p_i(T_w), i \in [0, 2]$ and the distribution of the packet size κ, we can define the 2-layered stochastic uncertainty model over time $\mathfrak{T}$ now. Referring to [72], this model represents an attempt to model the uncertainty of the TS itself and, since the probability $p_i(T_w)$ depends on the data rates $b(t)$ generated by $Model_B$, the stochastic uncertainty model can be defined in a similar way.

To this end, let $\mathfrak{M}$ be an instance of $Model_B$ and $\mathcal{N} = (\vec{X}_0, S, \mathcal{P}, \theta, \Lambda, \mathfrak{T}, \Sigma)$ be the network conditions of a communication scenario $\mathcal{C}$ as referring to Section 4.1. Then the instance $\mathfrak{M}$ can be transformed to an instance $\mathfrak{U}$ of an stochastic model describing the uncertainty of the underlying TS by replacing the system states $s(t) \in S_{\text{inner}}$ (radio link data rates) of the inner MC $\mathcal{Y}_{\text{inner}}$ of $\mathfrak{M}$ by probabilities p_0 and the patterns P_j representing the states of the outer MC by p_2. To this end, we define

```
Input: $C = (\mathcal{N},\mathcal{U})$, $\mathbf{P}_0$, $Msg$, $\kappa$, $t_{w_{max}}$, seed
Output: Optimum amount of redundancy $r$,
    maximized probability $\mathbb{P}(X \geq k|C)$
    Initialization :
$\Sigma \leftarrow Sample(\vec{X}_0$, $\mathfrak{B}$, $\theta$, $T$, $\lambda$, seed)
$S_{pt}, S_k \leftarrow []$
$p_{max}, r_{max} \leftarrow 0$
$t_w \leftarrow 1$
$k \leftarrow |\text{Msg}|$
while $t_w \leq t_{w_{max}}$ do
    for pattern $P_t \subseteq \mathcal{N}$ do
        for $k_t$ in $0, \dots, k$ do
            Given $k_t$ packets, compute the maximum amount of redundancy $r_t = \sum_{t=t_1}^{t_2} \epsilon \cdot b_{opt}(t)$ that the system can handle during $P_t = [b(t_1), \dots, b(t_2)]$
            Compute the probability $\mathbb{P}(X \geq k_t|P_t)$ with redundancy $r_t = n_t - k_t = \lfloor r_t/\kappa \rfloor$ by using $p_0 \in \mathbf{P}_0$
            $S_k$.append(($\mathbb{P}(X \geq k_t|P_t), r_t$))
        end for
        $S_{pt}$.append($S_k$)
    end for
    Compute the best configurations maximizing $p_{max} = \sum_{t=1}^{T}(\mathbb{P}(X \geq k_t|P_t))$ s.t. $\sum_{t=1}^{T} k_t \geq k$. Keep the solution guaranteeing $p_{max}$ with minimum amount of redundancy $r_{min} = \sum_{t=1}^{T} r_t = n_t - k_t$.
    if $p_{max} > p_{best}$ then
        $p_{best} \leftarrow p_{max}$
        $r_{best} \leftarrow r_{min}$
    end if
    $t_w + +$
end while
return $(p_{best}, r_{best})$
```

Algorithm 7 : MAXPROB [39]

the function ψ_{in} mapping each system state $s(t)$ of the inner MC to the probability p_0 by

$$\psi_{in} : (s(t), t) \to (\mathbb{P}(X = 1|b(t) = s(t), T_w(t), \kappa)\,. \tag{4.68}$$

Moreover, assuming that the outer MC $\mathcal{Y}_{\text{outer}}$ follows a pattern P_j generated by a single matrix $B_j \in P_j$, we can replace P_j by the maximized probability p_2 using ψ_{out} defined as

$$\psi_{out} : (P_j, t_1, t_2, k) \to \mathbb{P}\left(X \geq k|P_i = P_j, T_w(t_1, t_2), \kappa\right). \tag{4.69}$$

As a result, this model describes the distribution of uncertainty for packet delivery in the lower *layer 0* and the probability of message delivery in the higher *layer 2.*

5

Evaluation

This chapter evaluates the proposed models $Model_B$, $Model_{B^*}$ and $Model_{A|B}$ developed to solve both, *Problem B* and *Problem* $A|B$ using uniformly distributed user data flows. The experiments are generated using the theoretical frameworks described earlier in Chapter 4 and executed in a laboratory at Fraunhofer FKIE with VHF radios (Figure 5.1). It starts with experiments evaluating the basic performance metrics like latency, data rate and packet loss. Next, we discuss the TtR after unplanned link disconnections using the multi-agent $Model^*_B$. This is done by computing the inter-packet latency of three types of IP data-flows, namely broadcast, unicast and overlay. Finally, we report numerical calculations demonstrating how Model $A|B$ can compute close to optimal redundancy to improve the message/packet delivery over different loop scenarios with link disconnections. The experimental results also support the discussion on the advantages and limitations of our models, which were designed to test the robustness of tactical systems using military radios.

5.1 Experimental setup

The experimental setup used to compile the results in this thesis is composed two VHF radios (1) each connected to a router (2), who in turn is connected to a laptop (3), as illustrated in Figure 5.1. Each of the radios is equipped with a wired antenna and connected to a coaxial relays (0) in a laboratory environment. Moreover, the network setup has a relay, which is used to *"cut"* the antenna cable of a particular radio for a given time window, distributed with respect to λ (say 10 seconds) simulating link disconnection (state 0 of the models). All experiments that are considered in thins evaluation of the proposed models are communication scenarios consisting of ever-changing network conditions between the two nodes acting as *sender* (laptop 1) and *receiver* (laptop 2) over a uniform distributed user-data flow. More precise, the link data rate of the link connecting both nodes is changing according to the network conditions $\mathcal{N}$ of a communication scenario $\mathcal{C}$ generated by one of the models

$Model_B$ or $Model_{B^*}$, as referring to Section 4.1 and Section 4.2. During the experiments we collect all the logs from the nodes using a *data acquisition server* (4) and store the output data in a *Postgres* data base that can be accessed using a *Python* interface.

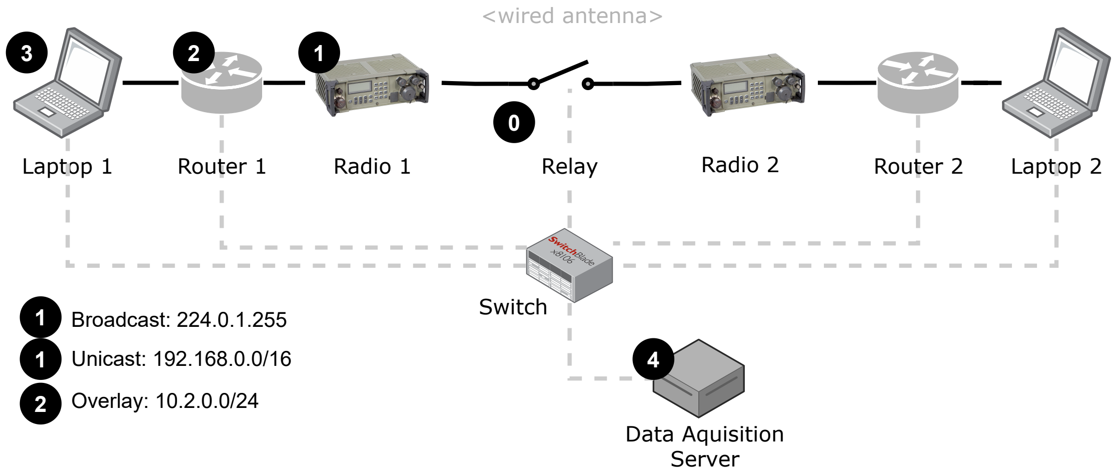


Figure 5.1 Network setup in the laboratory at Fraunhofer FKIE

Within the context of this study, we implemented the models $Model_B$, $Model^*_B$ and $Model_{A|B}$ (Chapter 4), as well as a script to control the radio modulation changing according to the network conditions of a respective communication scenarios using the radio's SNMP interface. The link disconnections are implemented by controlling the relay between the two radios used in the experiment. All experiments discuss several performance metrics of the tactical system with respect to the *broadcast/unicast/overlay* IP data-flows. To this end, the *sender* sends three types of IP packets using *broadcast/unicast/overlay* and we monitored the logs captured from both *sender* and *receiver*, as illustrated in Figure 5.1. The three IP data-flows are defined as follows:

○ *Broadcast*: allowing one-to-many data-flows by using the last IP in a domain, also supported by the VHF radios in our network setup at Fraunhofer FKIE. We used the IPs 192.168.1.255/24 or 192.168.255.255/16.

△ *Unicast*: only allows one-to-one data-flows that rely on the specific IP routes among the tactical nodes. In our network setup, the routes are configured statically among the tactical routers defining a fully connected topology;

+ *Overlay*: only allows one-to-one data-flows through an overlay IP network. In this network the routes are created dynamically by OLSR.

Table 5.1 summarizes the experimental setup listing the radio type and main configuration parameters used during the experiments, that we reported in this study.

5.2 Experiments using $Model_B$

In this section we discuss experimental results showing the three data flows *Broadcast*, *Unicast* and *Overlay* over the two loop scenarios L_1 and L_2. We also executed

Item	Description
Radio	4 VHF radios: PR4G by Thales with wired antennas
Time-window	10,40 and 180 seconds for state update
Data flows	broadcast, unicast and overlay
Network conditions	Communication scenarios: Generated by $Model_B$ and $Model_{B*}$
OLSRv2 [79]	15 and 45 seconds for *hello* and aggregation intervals

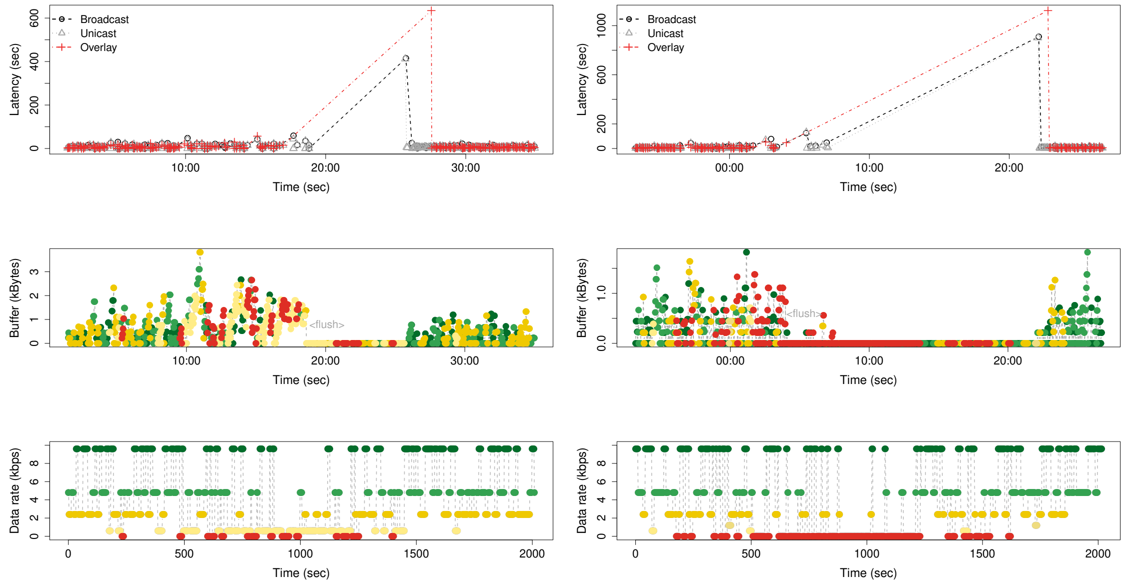
Table 5.1 Experimental setup

experiments for the two transformation (T_1 and T_2) and jump scenarios (J_1 and J_2). Since this thesis does not focus on $Model_B$ and basic performance metrics of tactical systems, but rather on the system's robustness, we refer to our publication [45] for a discussion of more elaborated results. The basic performance metrics plotted in this section are inter-packet latency (in seconds) measured at the *receiver* node, the buffer occupancy (in kBytes) and the link data rate (in kbps) observed at the *sender*. For both experiments, we set the state update interval for changing the network conditions according to the states of $Model_B$ to 10 seconds ($\lambda_t = 10$, $\forall t \in \mathfrak{T}$) and monitor the system behaviour for $|\mathfrak{T}| = 200$ states each, meaning that one experiment has a duration of 2000 seconds. If a period of link disconnections breaks one or more IP data-flows, we set the link to 4.8 kbps and wait until all broken data-flows are resumed as illustrated in Figure 5.2.

5.2.1 Experimental results

Comparing both loop scenarios L_1 and L_2, we observe that they differ in the network conditions created in the middle of the experiment, where L_2 shows higher frequency of link disconnections (Figure 5.2). Considering the performance metrics for the first loop as illustrated in Figure 5.2a, we observe a maximum latency of about 630 seconds (10.5 minutes). This is an indicator emphasizing that the scenario creates enough variation to restart the neighbor discovery for the *overlay*, meaning that the routes expired. In comparison, the second loop scenario L_2 has a maximum latency of about 1122 seconds, at least 200 seconds ($\sim$ 3.3 minutes), in the overlay being much higher than the values for *broadcast* and *unicast*, as illustrated in Figure 5.2b. As a result, the variation in the network conditions broke all three IP data flows after around 500 seconds of the experiment and the system was able to recover from the series of link disconnections after around 1700 seconds. Additionally, Table 5.2 lists the latency observed for the three data-flows (columns) during the two loop scenarios L_1 and L_2. This exposition of the very first performance metrics suggests that the *overlay* configuration is more susceptible to high variations in the network conditions, especially for scenarios with high frequency of link disconnections.

Next we consider the packet loss (in %) for the three IP data flows. Notice, that *broadcast* and *unicast* has significantly higher loss (30% and 34% for L_1 and 56% and 61% for L_2) than the *overlay* (14% for L_1 and 25% for L_2) for both scenarios. This might be logically explained by the fact that the dynamic routes expired during the series of link disconnection. As mentioned before, this means that the IP routes are deleted from the tactical system, thus not allowing the sender node to send packets to the radio. This is one main advantage of deploying an *overlay* network to the tactical system, since proactive neighbor discovery is capable of detecting the series of link disconnections and topology changes.



(a) Loop L_1 ~3.2 kbps ±3.1

(b) Loop L_2 ~3.9 kbps ±3.8

Figure 5.2 Latency(top), buffer(middle) and data rate(bottom) for two loops L_1 and L_2 [45]

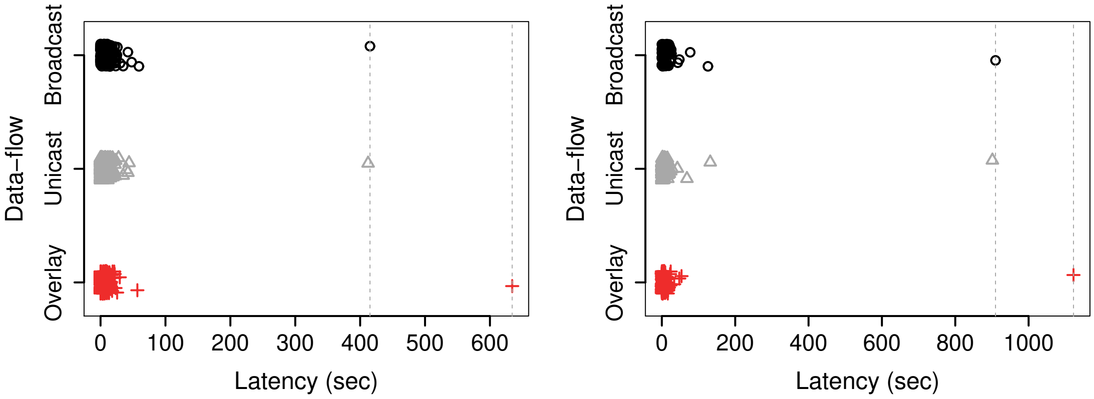
Pattern	Broadcast		Unicast		Overlay	
	Latency	Max	Latency	Max	Latency	Max
L_1 (Fig. 5.2a)	11.12 ± 31.5	415	7.56 ± 25.9	412	8.55 ± 41.6	634
L_2 (Fig. 5.2b)	19.43 ± 89.9	910	12.86 ± 72.7	901	14.42 ± 94.9	1122

Table 5.2 Average latency for three data-flows (columns) over two loop scenarios [45]

In addition, Figure 5.3 shows the range of inter-packet latency observed during both loop scenarios. These experimental results support the conclusion that the *overlay* data-flows have significantly higher latency if the network conditions of the scenario include slices with high frequency of link disconnections, as it is the case for L_2.

5.2.2 Summary

Discussing performance metrics like data rate, buffer occupancy, inter-packet latency and packet loss, we observed that the *overlay* is more susceptible to high variations in the network conditions. But, even if the neighbour discovery of the *overlay* configuration is restarted more frequently as the *broadcast* and *unicast* data-flows are broken, deploying an *overlay* network has the advantage of proactive neighbor discovery, which is capable of detecting series of link disconnections and topology changes, thus reducing packet loss. For communication scenarios with high frequency of link disconnections (scenario L_2) we saw that the network conditions generated by $Model_B$ were able to break all three IP data-flows. In this scenarios, the *overlay* data-flows shows significantly higher latency, but for *broadcast* and *unicast* we observed higher packet loss. We conclude, that $Model_B$ is capable of creating ever-changing network conditions that can be used to analyze the performance bounds of tactical systems. But, creating these network conditions is linked to the problem of defining transformation functions that can “tame the randomness” of the



(a) Loop scenario L_1

(b) Loop scenario L_2

Figure 5.3 Inter-packet latency at the *receiver* [45]

in-homogeneous Markov chain, which requires precise knowledge about the underlying stochastic model. Also, there is no mechanism to automate the experiment generation process by including system feedback directly in the modeling process. Keeping this limitations of $Model_B$ in mind, we will discuss the results and benefits of $Model_{B*}$ and $Model_{A|B}$ in the following sections.

$$
\bar{B}_0\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5\\ 0 & .67 & .33 & 0 & 0 & 0 & 0\\ 1 & .67 & .33 & 0 & 0 & 0 & 0\\ 2 & .67 & .33 & 0 & 0 & 0 & 0\\ 3 & .67 & .33 & 0 & 0 & 0 & 0\\ 4 & .67 & .33 & 0 & 0 & 0 & 0\\ 5 & .67 & .33 & 0 & 0 & 0 & 0 \end{array}
\bar{B}_1\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5\\ 0 & 0 & .67 & .33 & 0 & 0 & 0\\ 1 & 0 & .67 & .33 & 0 & 0 & 0\\ 2 & 0 & .67 & .33 & 0 & 0 & 0\\ 3 & 0 & .67 & .33 & 0 & 0 & 0\\ 4 & 0 & .67 & .33 & 0 & 0 & 0\\ 5 & 0 & .67 & .33 & 0 & 0 & 0 \end{array}
\bar{B}_2\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5\\ 0 & 0 & .0002 & .6698 & .33 & 0 & 0\\ 1 & 0 & .0002 & .6698 & .33 & 0 & 0\\ 2 & 0 & .0002 & .6698 & .33 & 0 & 0\\ 3 & 0 & .0002 & .6698 & .33 & 0 & 0\\ 4 & 0 & .0002 & .6698 & .33 & 0 & 0\\ 5 & 0 & .0002 & .6698 & .33 & 0 & 0 \end{array}
$$

$$
\bar{B}_3\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5\\ 0 & 0 & 0 & .1 & .8995 & .0005 & 0\\ 1 & 0 & 0 & .1 & .8995 & .0005 & 0\\ 2 & 0 & 0 & .1 & .8995 & .0005 & 0\\ 3 & 0 & 0 & .1 & .8995 & .0005 & 0\\ 4 & 0 & 0 & .1 & .8995 & .0005 & 0\\ 5 & 0 & 0 & .1 & .8995 & .0005 & 0 \end{array}
\bar{B}_4\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5\\ 0 & 0 & 0 & 0 & .1 & .9 & 0\\ 1 & 0 & 0 & 0 & .1 & .9 & 0\\ 2 & 0 & 0 & 0 & .1 & .9 & 0\\ 3 & 0 & 0 & 0 & .1 & .9 & 0\\ 4 & 0 & 0 & 0 & .1 & .9 & 0\\ 5 & 0 & 0 & 0 & .1 & .9 & 0 \end{array}
\bar{B}_5\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5\\ 0 & 0 & 0 & 0 & 0 & .1 & .9\\ 1 & 0 & 0 & 0 & 0 & .1 & .9\\ 2 & 0 & 0 & 0 & 0 & .1 & .9\\ 3 & 0 & 0 & 0 & 0 & .1 & .9\\ 4 & 0 & 0 & 0 & 0 & .1 & .9\\ 5 & 0 & 0 & 0 & 0 & .1 & .9 \end{array}
\tag{5.1}
$$

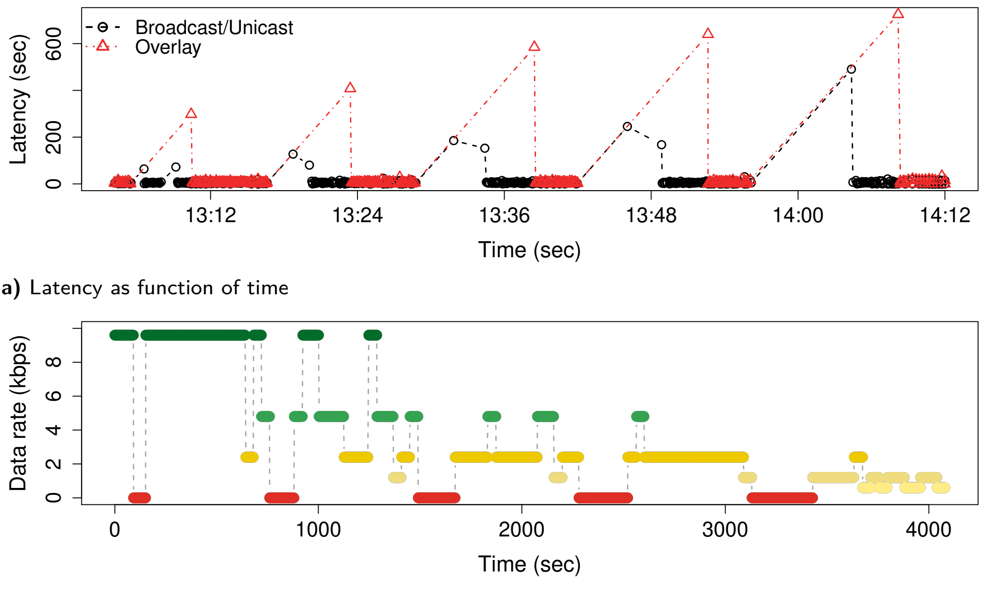
5.3 Experiments using *Model*$_{B^*}$

In this section, we discuss the results related to the multi-agent Model$_{B^*}$. Remind that the idea was to enhance *Model*$_B$ to a model that can automatically generate a set of experiments to learn a predefined target metric. In this thesis, the model was deployed to learn the distribution of the TtR after unplanned link disconnections, which represent the worst event in any communication scenario. To this end, we instantiated the first agent to solve a MDP, thus creating the ever-changing network conditions of communication scenarios including link disconnections varying from 60, 120, ... to 300 seconds. We monitor the data captured from the different layers of the tactical system during each experiment and perform different statistical tests analyzing the EDF of the TtR. Then a second agent decides if the distribution TtR converges in distribution, independently of the duration of the link disconnection. As a result, *Model*$_B$ defines a solution to *Problem B* by learning the TtR, which is a metric describing the robustness of tactical systems for worst-case events (unplanned link disconnections). The experiments discussed in this section use exemplary data flows composed of Internet Control Message Protocol (ICMP) packets (~300 Bytes) that are send in a rate of 5 seconds using three different IPs in the network, namely *broadcast*, *unicast* and *overlay.* In our laboratory environment (Figure 5.1), *broadcast* and *unicast* define the basic IP functionalities supported by our radios (PR4G by Thales), and the overlay network was created using OLSRv2. We tested two different configurations for the neighbor discovery (say interval of 45 and 15 seconds for *hello* and *topology control* messages) of the *overlay* network. Therefore, the *overlay* network depends on *broadcast* to compile the network topology using proactive neighbor discovery. This dependency is visible in the quantitative results discussed in the following section.

5.3.1 Experimental results

Let us start the discussion with compiling some baseline results from an experiment without link disconnections, as illustrated in Figure 5.4b. The experiment was generated by sampling a sequence of data rates from the optimal policy $v^* = [\bar{B}_1, \bar{B}_2, \bar{B}_3, \bar{B}_4, \bar{B}_5]$ defined using the probability distributions as shown in Figure 5.1. Complementing, Figure 5.4a represents the latency observed during the experiment.

Notice that higher link data rates resulted in clearly lower latency times, demonstrated as the upper plot (Figure 5.4a) fluctuates a lot less as the data rate in the plot below Figure 5.4b approaches best conditions (dark green, 9,6kbps). This supports the results in Section 5.2.1 from $Model_B$ and shows that $Model_{B^*}$ can be used to compile similar results as $Model_B$.



(a) Latency as function of time

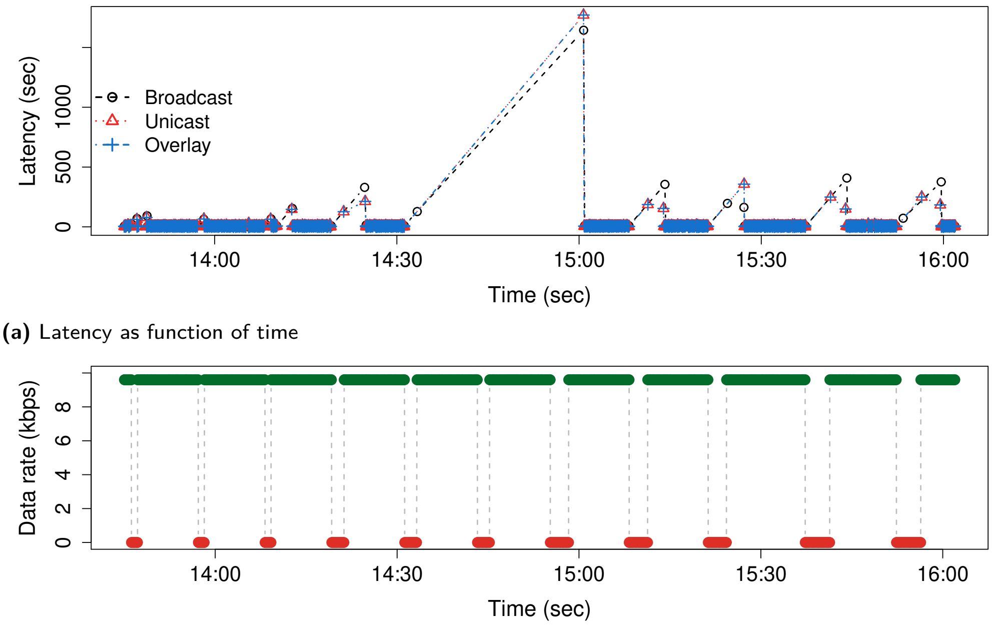
(b) Data rate as function of time

Figure 5.4 Experiments without link disconnections

Under these conditions, we moved on creating experiments varying the link disconnection time using agent A_{MDP}. Figure 5.5 illustrates the behaviour of agent A_{MDP} increasing the link disconnection time each time agent A_{H_0} verified that the distribution of the TtR met the null hypothesis locally. The first thing to be noticed was that we need at least 60 seconds of link disconnection to guarantee that all three IP data-flows are broken. Thus, we decided to create communication scenarios including link disconnections varying from 60, 120, ... to 300 seconds.

Figure 5.6 shows another scenario moving from best conditions (9.6kbps) to bad conditions (0.6kbps) in between sequences of increasing link disconnections. In this figure we plot the inter-packet latency (top plot) over the changes in the link data rate (middle plot) and the TtR (bottom plot) as a function of time. As for the simple scenario, only jumping between best and worst conditions (Figure 5.5), the peak of latency occurs after the link disconnections verifying that unplanned link disconnections are the worst-case event during the communication scenario. Moreover, we can observe that the *overlay* data flow needs a longer time to recover from worst-case events than *broadcast* and *unicast*. In the following we will discuss how to determine the minimum TtR for each of the IP data flows.

To this end, we initialized agent A_{MDP} with a MDP whose optimal policy represents a Markov chain jumping between best and worst conditions as shown in Figure 5.5. We decided for this simple setup, since the minimum TtR should be discovered using best conditions after each link connection, thus allowing the system to use maximum capacity to recover from the broken link.



(a) Latency as function of time

(b) Data rate as function of time

Figure 5.5 Experiments varying the link disconnection time

Figure 5.7a shows a box-plot of the EDF of the TtR for *broadcast/unicast*, which represents the baseline results for the discussion. For 60 seconds disconnections the distribution of the TtR is defined by a function fluctuating around 220 seconds. But, for longer disconnections, starting at 120 seconds, we see more variation in the TtR, meaning that the distributions have a median between 200 and 400 seconds and we also observe peak values close to 900 seconds. In addition, Figure 5.7b shows an equivalent plot illustrating that the *overlay* adds an additional overhead to the TtR, and it depends on the neighbor discovery parameters, such as *hello interval* and *topology control*. We conclude this from the fact that we observe much higher peak values of the TtR (around 1300 seconds) for the overlay network, while the median of the EDF remains around 400 seconds.

As mentioned in Section 4.2.2.5, agent A_{H_0} performs three different statistical tests to verify if the null hypothesis can be accepted locally/globally within a confidence interval of 95%. The results from these statistical tests for *broadcast/unicast* are shown in Figure 5.7c and also listed in Table 5.3. We observe that for disconnections greater than 60 and smaller than 300 seconds, the statistical tests return great similarity. This is because we observe high p-values by pairwise comparing the EDF of the TtR for the K-S test, meaning that the K-S test statistics are lower than the confidence threshold, also highlighted *green* in Table 5.3. Moreover, the pairwise normalized Wasserstein distance supports this result being lower or equal to 0.2.

Figure 5.7d shows mostly similar results for the EDFs of the TtR in the *overlay* network illustrating that the distribution starts converging at comparison 2 of 120 and 180 seconds. This is indicated by the p-value and the statistics of the K-S test, the first one being close to 1 and the second one being lower than the confidence threshold. Comparing the distributions for a disconnection time of 180 to 240 seconds the p-value reaches a value around 1.6%. In this case, the K-S test test suggests to reject the null hypothesis indicating that the distributions are slightly different.

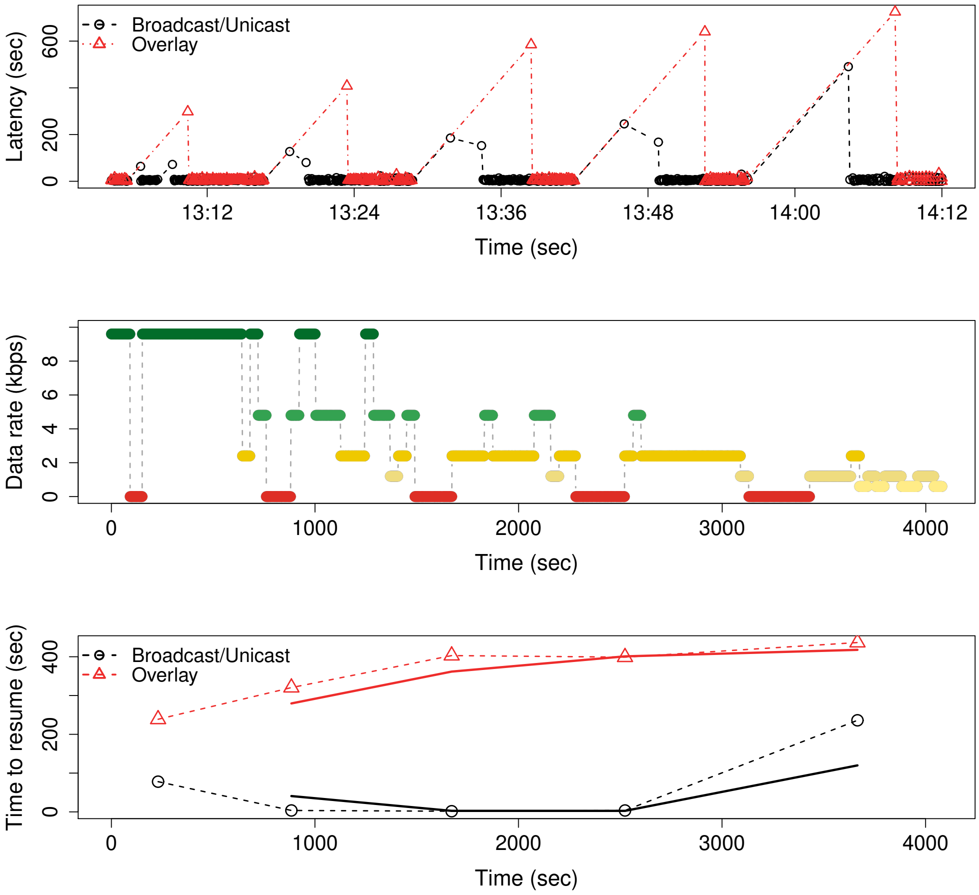
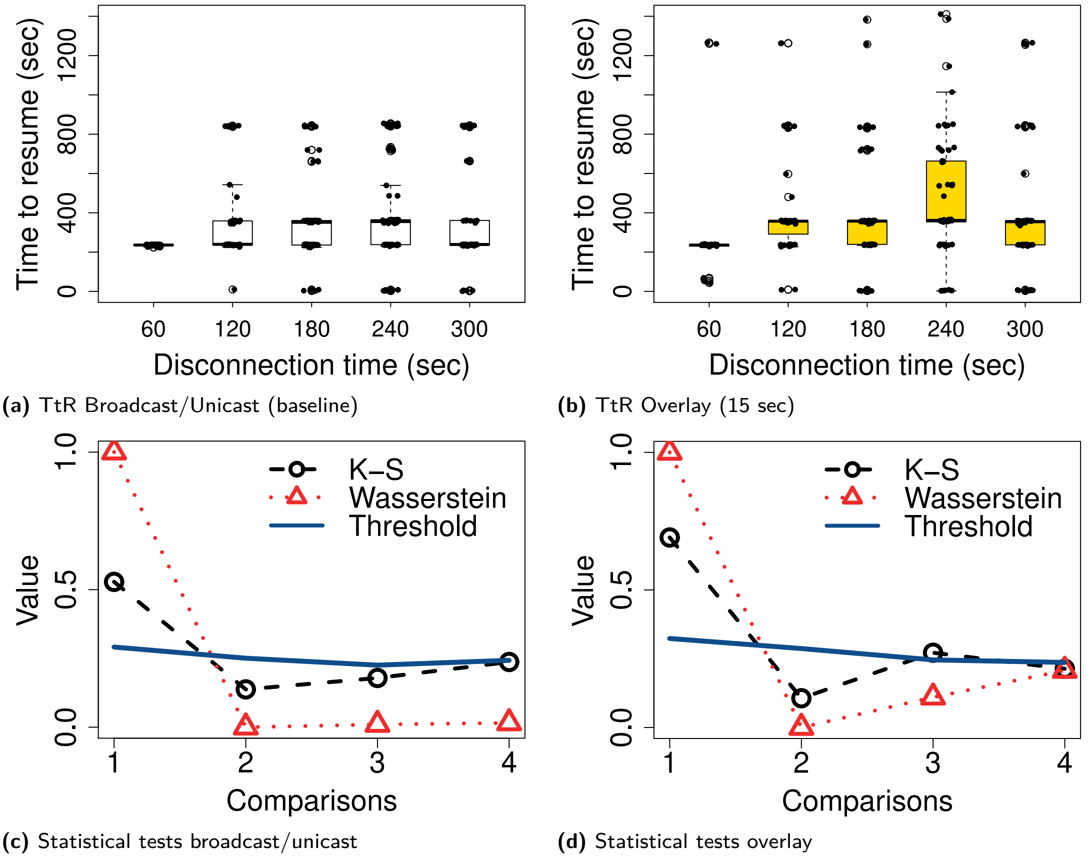


Figure 5.6 Experiments with different link disconnection time [44]

IP data-flow	Comparison	KS-statistics	Threshold	p-Value	Wasserstein distance (normalized)
Broadcast/Unicast	1	0.53	0.29	0.0	1.0
	2	0.14	0.25	0.57	0.0
	3	0.18	0.22	0.17	0.0
	4	0.23	0.24	0.14	0.0
Overlay 15s	1	0.69	0.32	0.0	1.0
	2	0.11	0.29	0.92	0.0
	3	0.27	0.24	0.016	0.11
	4	0.21	0.24	0.82	0.2
Overlay 45s	1	0.40	0.35	0.01	0.77
	2	0.31	0.36	0.2	0.16
	3	0.14	0.41	0.89	0.0
	4	0.31	0.41	0.2	0.08

Table 5.3 Results from the statistical tests for *broadcast/unicast* and *overlay*

But, the Wasserstein distance being around 0.11, as listed in Table 5.3 for *overlay* 15s, remains lower than 0.2 and since this test can also handle small perturbations in probability distributions, this metric defines the most important metric for the agent deciding that the distributions are similar. Moving from comparison 3 to 4 the situation relaxes, meaning that the CDF of TtR is still converging. As a result, the second agent went to state 2 *"overall distribution meets null hypothesis"* and stopped the experiment.



(a) TtR Broadcast/Unicast (baseline)

(b) TtR Overlay (15 sec)

(c) Statistical tests broadcast/unicast

(d) Statistical tests overlay

Figure 5.7 Distribution of the TtR as function of the disconnection time [44]

5.3.2 Summary

In summary, we conclude that the multi-agent model $Model_{B^*}$ is capable of learning the CDF of the TtR in worst-case scenarios (unplanned link disconnections). To this end, the model automates the experiment generation process using two agents and performs three statistical tests (Section 4.2.2.5) for each experiment to verify that the null hypothesis can be accepted locally and globally, meaning that the distribution for the TtR is similar and independent of the disconnection time. As a result, we got a first performance metric measuring the robustness of TS for worst-case events. Nevertheless, it remains an open issue how to adapt the model to learn arbitrary performance metrics of TS during arbitrary communication scenarios, also in such cases, where the performance metric previously may be unknown to the human observer.

5.4 Experiments using *Model*$_{A|B}$

This section discusses experimental results of the stochastic uncertainty $Model_{A|B}$ (Section 4.3.2) observed during two end-to-end communication scenarios $\mathcal{C}_{L_1} = (\mathcal{N}_1, \mathcal{U}_1)$ and $\mathcal{C}_{L_2} = (\mathcal{N}_2, \mathcal{U}_2)$, such that the network conditions are defined by the loops $\mathcal{N}_1 = L_1$ and $\mathcal{N}_2 = L_2$. The user-data flows $\mathcal{U}_1 = \mathcal{U}_2$ are given by messages composed by 1, 3, 9, 18 and 54 packets, and the sender packet rate was distributed according to $(0.05, 0.075, 0.1, 0.2, 0.5, 1, 2)$ *packet/second*. As well as for $Model_B$ and

C	*r(t)*	**Number of packets per message**				
		1	3	9	18	54
L_1^*	0%	.69 (±.11)	.33 (±.08)	.04 (±.01)	.00 (±.00)	.00 (±.00)
L_2^*		.63 (±.09)	.25 (±.05)	.02 (±.00)	.00 (±.00)	.00 (±.00)
L_1	0%	.49 (±.11)	.30 (±.08)	.04 (±.01)	.00 (±.00)	.00 (±.00)
L_2		.61 (±.09)	.25 (±.05)	.02 (±.00)	.00 (±.00)	.00 (±.00)
L_1	100%	.69 (±.12)	.90 (±.17)	.97 (±.21)	.99 (±.24)	.99 (±.26)
L_2		.71 (±.11)	.86 (±.14)	.92 (±.18)	.96 (±.20)	.99 (±.22)
L_1	200%	.82 (±.12)	.99 (±.14)	.99 (±.16)	.99 (±.16)	.99 (±.16)
L_2		.76 (±.11)	.98 (±.14)	.99 (±.15)	.99 (±.15)	.99 (±.16)
L_1	500%	.96 (±.10)	.99 (±.12)	1.0 (±.12)	1.0 (±.13)	1.0 (±.13)
L_2		.97 (±.10)	.99 (±.11)	1.0 (±.12)	1.0 (±.12)	1.0 (±.13)

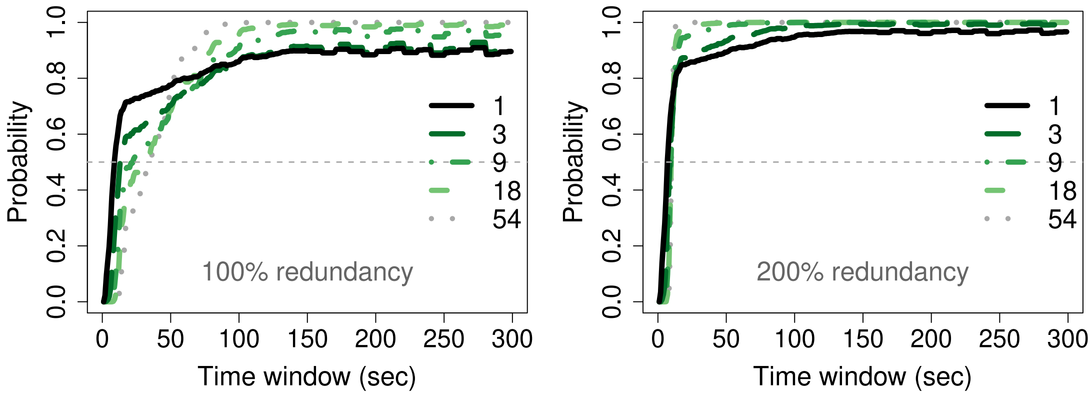
Table 5.4 Probability of message delivery (0.2 packet/second) [39].

$Model_{B^*}$, we use the VHF network from the experimental setup shown in Figure 5.1. In comparison to $Model_{B^*}$, which focuses on the worst-case of unplanned link disconnections, $Model_{A|B}$ can be used to measure and improve the robustness of tactical systems for arbitrary communication scenarios. Even if the results discussed in this section focus on two scenarios generated using $Model_B$, the model can be applied for any other communication scenario, also generated by $Model_B$. The only requirement for applying $Model_{A|B}$ is, that the model has access to the different layers of a tactical system as illustrated earlier in Figure 3.1.

5.4.1 Experimental results

Both communication scenarios L_1 and L_2 include very low link data rates (i.e 0.6 and 1.2 kbps) and even link disconnections (0 kbps) resulting in packet loss. More precisely, L_1 (Figure 4.6a) represents a loop of 200 states updated every 10 seconds and morphing a pattern distributed according to B_4 into a pattern which is distributed according to B_1 and back to B_4. In comparison, Figure 4.6b shows the loop L_2, which is generated by replacing the pattern distributed by B_1 by $\hat{B}_0$. Since $\hat{B}_0$ has a higher probability for link disconnections, the pattern shows higher frequency of link disconnections in the middle of the communication scenario. As a result, we observe increased packet loss of 54% for L_2 compared to L_1, where the loss is only 5%. We have chosen these two patterns because they concentrate the disconnections in the middle of the experiment, meaning that we can see the system recovering after the series of disconnections.

To evaluate the output of $Model_{A|B}$, we compare the probabilities p_2 computed by Algorithm 7 assuming that the ground truth p_0 for the respective communication scenario is known before with the probabilities p_2 computed by using the ground truth probabilities measured under stable system conditions. Thus, the first case represents a baseline and will be used to demonstrate that Algorithm 7 can also compute reliable output without any prior knowledge about the communication scenario. The latter case represents a typical configuration if the model is deployed to a tactical system, meaning that it has to react to the ever-changing network conditions in time.



.

Figure 5.8 Probability as function of the time window [39]

Table 5.4 illustrates the results for four levels of redundancy 0, 100, 200 and 500 % and for different maximum end-to-end delay $|T_w| \in \{1, \dots, 300\}$, assuming that the packet sending rate is 0.2 *packet/second*. In this table, the star indicates that L_1^* and L_2^* are the baseline results computed assuming that the ground truth p_0 is known. Notice that, for both computations of the probability of message delivery p_2 L_1^*, L_2^* (lines 1 and 2 in the table) and L_1 and L_2 (lines 3 and 4 in the table) without redundancy, $r(t) = 0\%$, the probability of message delivery is about 60% for messages with 1 packet and goes decreasing until 0% for messages with 18 or more packets. If we start increasing the amount of redundancy to 100% , we observe that adding redundancy can significantly increase the probability of message delivery for both loop scenarios. For messages consisting of a single packet, we need up to 500% of redundancy to guarantee reliable message delivery. For messages consisting of 3 or more packets, we see that even 100% redundancy can lead to satisfying results (i.e. probabilities close to 1).

Figure 5.8 complements the results from this table by showing the probability for message delivery as a function of different time windows T_w for six messages sizes, namely 1, 3, 9, 18 and 54 packets per message. In this figure, there are two examples representing 100% (left) and 200% (right) for L_1 showing that the probabilities converge with respect to the size of the maximum end-to-end delay $|T_w|$; also called time window. This is an important result, because it yields to the assumption that we can replace the parameter $t_{w_{max}}$ by another parameter describing the convergence rate of Algorithm 7.

5.4.2 Summary

The experimental results showed that $Model_{A|B}$ is able to increase the probability of message delivery in ever-changing communication scenarios significantly. Moreover, the model only depends on the probabilities for the ground truth and can be applied to arbitrary communication scenarios. As a result, $Model_{A|B}$ can be used to measure and improve the robustness of tactical systems in ever-changing communication scenarios generated by $Model_B$ or $Model_{B^*}$. But, we notice that for communication scenarios with long periods of disconnections as L_2, there is a higher demand of redundancy. The reason is that the model is not capable of recognizing and adapting to the period of disconnection. In contrast, it fills the packet queues with redundancy packets and attempts to solve the problem by duplicating enough packets to overcome the packet loss in the middle of the scenario. As a result, the robustness of the tactical systems in communication scenarios with long periods of disconnections can only be improved by adding a high amount of redundancy, which adds an additional overhead to the system that is already fighting with low data rates and high latency. In other words, the system is missing the ability to recognize periods of link disconnections and to adapt to these scenarios by exploiting the knowledge of the minimum TtR after link disconnections as discussed earlier in Section 4.2.2.4.

6

Conclusion

This thesis introduced three stochastic models aiming to measure and finally improve the robustness of TS by using cross-layer information gathered from the hierarchical layers, which form the TS. The goal was to introduce a reproducible mechanism to quantify if the TS can handle the user data flow given the circumstances arising from ever-changing tactical situations and environmental conditions on the battlefield (ever-changing network conditions). We started with the hypothesis that we can develop a model quantizing the network conditions using a discrete set of states representing the radio modulations supported by tactical radios, thus defining the distribution of change in the link connection. This model was enhanced to a MDP afterwards, allowing to introduce system feedback directly into the modeling process. Moreover, the MDP was used to automate the experiment process by defining a multi-agent system to measure the system robustness by computing the distribution of the TtR after unplanned link disconnections. Finally, we introduced a stochastic uncertainty model to improve the system robustness by proactively adding redundancy, thus maximizing the probability of message/packet delivery using the hierarchy of layers from modern TSs. The main idea was to estimate the optimum redundancy level by solving an optimization problem to mitigate packet loss in communication scenarios with link disconnections, therefore increasing the probability of delivering messages. We evaluated the three models over a variety of different experiments executed in a laboratory environment at Fraunhofer FKIE with real military radios. The experimental results suggest that our stochastic model can compute close to optimal parameters for a transport protocol using redundancy to overcome packet loss during link disconnections, also avoiding data overhead from packet acknowledgements and packet re-transmissions. Even if the models introduced in this thesis are developed at the edge of TNs, but nevertheless they meet the requirements of generalizability, such that they can be applied to arbitrary transport protocols to proactively add redundancy to reduce the packet loss observed during radio link disconnections.

6.1 Publications

The publications related to the present thesis are as follows:

- Development of the stochastic uncertainty $\mathit{Model}_{A|B}$ as referring to Section 4.3:

  Loevenich, J. F., Lopes, R. R. F., Rettore, P. H., Eswarappa, S. M., and Sevenich, P. Maximizing the probability of message delivery over ever-changing communication scenarios in tactical networks. *IEEE Networking Letter 3*, 2 (June 2021), 84–88

- Development of the multi-agent Model_{B*} as referring to Section 4.2:

  Lopes, R. R. F., Loevenich, J., Sergeev, A., Rettore, P. H., and Sevenich, P. Quantifying the robustness to unplanned link disconnections in tactical networks. In *IEEE Conference on Military Communications (MILCOM)* (San Diego, USA, Nov. 2021). (under review)

- Development of the stochastic Model_B as referring to Section 4.1:

  Lopes, R. R. F., Loevenich, J. F., Rettore, P. H., Eswarappa, S. M., and Sevenich, P. Quantizing radio link data rates to create ever-changing network conditions in tactical networks. *IEEE Access 8* (September 2020), 188015–188035

- Development of a test platform to test military systems used to evaluate the stochastic models (Section 5):

  Rettore, P. H., Loevenich, J., Lopes, R. R. F., and Sevenich, P. TNT: A tactical network test platform to evaluate military systems over ever-changing scenarios. *IEEE/ACM Trans. Netw.* (Mar. 2021). pre-print, doi:10.36227/techrxiv.14141501

Other parallel investigations with contributions from Mr. Loevenich during his internship as a master student at Fraunhofer FKIE:

- Balaraju, P. H., Rettore, P. H., Lopes, R. R. F., Eswarappa, S. M., and Loevenich, J. Dynamic adaptation of the user data flow to the changing data rates in VHF networks: An exploratory study. In *11th IEEE International Conference on Network of the Future (NoF)* (Bordeaux, France, Oct 2020), pp. 64–72

- Eswarappa, S. M., Rettore, P. H. L., Loevenich, J., Sevenich, P., and Lopes, R. R. F. Towards adaptive qos in sdn-enabled heterogeneous tactical networks. In *IEEE International Conference on Military Communication and Information Systems (ICMCIS)* (2021), pp. 1–8

- Lopes, R. R. F., Rettore, P. H., Eswarappa, S. M., Loevenich, J., and Sevenich, P. Performance analysis of proactive neighbor discovery in a heterogeneous tactical network. In *IEEE International Conference on Military Communication and Information Systems (ICMCIS)* (2021), pp. 1–8

- Rettore, P. H., von Rechenberg, M., Loevenich, J., Rigolin F. Lopes, R., and Sevenich, P. A handover mechanism for Centralized/Decentralized networks over disruptive scenarios. In *MILCOM 2021 Track 2 - Networking Protocols and Performance (MILCOM 2021 Track 2)* (San Diego, USA, nov 2021). (under review)

6.2 Future work

Recent literature reported intelligent systems learning to play games from scratch without human supervision [73]. To this end, the intelligent system combines different learning strategies to discover the problem space and learn metrics that are not known to the human developer before. Inspired by this approach, we think that the multi-agent model introduced here can be extended to learn several performance metrics without human intervention. The general idea to enhance $Model_{B^*}$ to such a system will be to remove the target metric controlling the MDP and introduce the TS parameters directly into the underlying Bellman equation of the stochastic process by using physical equations. Then the goal of the multi-agent system will be to find scenarios breaking the configuration of TS, thus defining the performance bounds of the current configuration and to reestablish reliable communication by optimizing the parameters of the respective stochastic optimization problem. The resulting communication scenarios and performance metrics will then be stored in a database accessible to the TS and can be used to find the optimal configuration in online situations by comparing the current scenario (using the Wasserstein metric 4.2) to the knowledge defined by the scenarios in the database. Even if, this is a very ambitious goal, which requires expert knowledge in learning strategies, physics, and mathematics we think that this can be topic for a future Ph.D. thesis.

Bibliography

[1] Ahrenholz, J., Danilov, C., Henderson, T. R., and Kim, J. H. CORE: A real-time network emulator. In *IEEE Military Communications Conference (MILCOM)* (2008), pp. 1–7.

[2] Altschuler, J., Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration, 2018.

[3] Balaraju, P. H., Rettore, P. H., Lopes, R. R. F., Eswarappa, S. M., and Loevenich, J. Dynamic adaptation of the user data flow to the changing data rates in VHF networks: An exploratory study. In *11th IEEE International Conference on Network of the Future (NoF)* (Bordeaux, France, Oct 2020), pp. 64–72.

[4] Bellman, R. A markovian decision process. *Indiana Univ. Math. J. 6* (1957), 679–684.

[5] Bellman, R. E. *Dynamic Programming.* Dover Publications, Inc., USA, 2003.

[6] Benamou, J., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM J. Sci. Comput. 37* (2015).

[7] Benincasa, G., Bunch, L., Casini, E., Lenzi, R., Morelli, A., Paulini, M. S., Suri, N., and Uszok, A. Bridging the gap between enterprise and tactical networks via mission- and network-sensitive adaptation. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2018), pp. 1–8.

[8] Blondel, M., Seguy, V., and Rolet, A. Smooth and sparse optimal transport, 2018.

[9] Boyan, J. A., and Littman, M. L. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Proceedings of the 6th International Conference on Neural Information Processing Systems* (San Francisco, CA, USA, 1993), NIPS'93, Morgan Kaufmann Publishers Inc., p. 671–678.

[10] Carl, R., Swanson, K., Bonney, J., and Trent, B. Transport protocols in the tactical network environment. In *2007 IEEE Aerospace Conference* (2007), pp. 1–9.

[11] Casini, E., Benincasa, G., Morelli, A., Suri, N., and Breedy, M. An experimental evaluation of data distribution applications in tactical networks. In *IEEE Military Communications Conference (MILCOM)* (Nov 2016), pp. 1267–1272.

[12] Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. Scaling algorithms for unbalanced transport problems, 2017.

[13] Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.

[14] Dalkiran, E., Önel, T., Topçu, O., and Demir, K. A. Automated integration of real-time and non-real-time defense systems. *Defence Technology* (2020).

[15] Danilov, C., Henderson, T. R., Goff, T., Brewer, O., Kim, J. H., Macker, J., and Adamson, B. Adaptive routing for tactical communications. In *IEEE Military Communications Conference (MILCOM)* (2012), pp. 1–7.

[16] Diefenbach, A., Lopes, R. R. F., Lampe, T. A., Prasse, C., Śliwa, J., Goniacz, R., and Viidanoja, A. Realizing overlay Xcast in a tactical service infrastructure: An approach based on a service-oriented architecture. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2018), pp. 1–8.

[17] Dvoretzky, A., Kiefer, J., and Wolfowitz, J. Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *The Annals of Mathematical Statistics 27*, 3 (1956), 642 – 669. https://doi.org/10.1214/aoms/1177728174.

[18] Eswarappa, S. M., Rettore, P. H. L., Loevenich, J., Sevenich, P., and Lopes, R. R. F. Towards adaptive qos in sdn-enabled heterogeneous tactical networks. In *IEEE International Conference on Military Communication and Information Systems (ICMCIS)* (2021), pp. 1–8.

[19] Evans, J., Ewy, B., Swink, M., Pennington, S., Siquieros, D., and Earp, S. TIGR: the tactical ground reporting system. *IEEE Communications Magazine 51*, 10 (October 2013), 42–49.

[20] Frogner, C., Zhang, C., Mobahi, H., Araya-Polo, M., and Poggio, T. Learning with a wasserstein loss, 2015.

[21] Fronteddu, R., Morelli, A., Mantovani, M., Ordway, B., Campioni, L., Suri, N., and Marcus, K. M. State estimation for tactical networks: Challenges and approaches. In *IEEE Military Communications Conference (MILCOM)* (Oct 2018), pp. 1042–1048.

[22] Fronteddu, R., Morelli, A., Tortonesi, M., Suri, N., Stefanelli, C., Lenzi, R., and Casini, E. DDAM: Dynamic network condition detection and communication adaptation in tactical edge networks. In *IEEE Military Communications Conference (MILCOM)* (Nov 2016), pp. 970–975.

[23] Gelb, M. J. *How to Think Like Leonardo da Vinci.* Dell, Berlin, Heidelberg, 2000.

[24] Ghosh, A., Li, S., Chiang, C. J., Chadha, R., Moeltner, K., Ali, S., Kumar, Y., and Bauer, R. QoS-aware adaptive middleware (QAM) for tactical MANET applications. In *IEEE Military Communications Conference (MILCOM)* (Oct 2010), pp. 178–183.

[25] Grushevsky, Y. L., and Elmasry, G. F. Adaptive RS codes for message delivery over an encrypted mobile network. *IET Communications 3*, 6 (2009), 1041–1049.

[26] Hannay, J. Architectural work for modeling and simulation combining the NATO architecture framework and C3 taxonomy. 139–158.

[27] Hannay, J. Architectural work for modeling and simulation combining the nato architecture framework and C3 taxonomy. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology 14* (11 2016).

[28] Howard, R. A. Cambridge, MA.

[29] Häggström, O. *Finite Markov Chains and Algorithmic Applications.* London Mathematical Society Student Texts. Cambridge University Press, 2002.

[30] IEEE. IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990* (Dec 1990), 1–84.

[31] Johnsen, F. T., Bloebaum, T. H., Calero, J. M. A., Wang, Q., Nightingale, J., Manso, M., and Jansen, N. WS-notification case study and experiment. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2017), pp. 1–8.

[32] Johnston, M., Danilov, C., and Larson, K. A reinforcement learning approach to adaptive redundancy for routing in tactical networks. In *IEEE Military Communications Conference (MILCOM)* (2018), pp. 267–272.

[33] Kantorovich, L. On the translocation of masses. *Journal of Mathematical Sciences 133* (2006), 1381–1382.

[34] Kaviani, S., Ryu, B., Ahmed, E., Larson, K. A., Le, A., Yahja, A., and Kim, J. H. Robust and scalable routing with multi-agent deep reinforcement learning for manets, 2021.

[35] Kumar, S., and Miikkulainen, R. Confidence-based q-routing: An online adaptive network routing algorithm. In *In Proceedings of Artificial Neural Networks in Engineering* (1998).

[36] Lee, S., Choi, J. Y., Lim, K. W., Ko, Y., and Roh, B. A reliable and hybrid multi-path routing protocol for multi-interface tactical ad hoc networks. In *IEEE Military Communications Conference (MILCOM)* (2010), pp. 2237–2242.

[37] Lindquister, J. J., Johnsen, F. T., and Bloebaum, T. H. Proxy pair optimizations for increased service reliability in DIL networks. In *IEEE Military Communications Conference (MILCOM)* (Oct 2017).

[38] Locher, J., and Veldhuysen, W. *The Magic of M.C. Escher.* Thames & Hudson, New York, NY, USA, 2013.

[39] Loevenich, J. F., Lopes, R. R. F., Rettore, P. H., Eswarappa, S. M., and Sevenich, P. Maximizing the probability of message delivery over ever-changing communication scenarios in tactical networks. *IEEE Networking Letter 3*, 2 (June 2021), 84–88.

[40] Lopes, R. R. F., Balaraju, P. H., and P.Sevenich. Creating ever-changing QoS-constrained dataflows in tactical networks: An exploratory study. In *International Conference on Military Communications and Information Systems (ICMCIS)* (Budva, Montenegro, May 2019), pp. 1–8.

[41] Lopes, R. R. F., Balaraju, P. H., Rettore, P. H., and Sevenich, P. Queuing over ever-changing communication scenarios in tactical networks. *IEEE Transactions on Mobile Computing* (June 2020), 1–15. (early access).

[42] Lopes, R. R. F., Balaraju, P. H., and Sevenich, P. Creating and handling ever-changing communication scenarios in tactical networks. In *15th International Conference on the Design of Reliable Communication Networks (DRCN)* (Coimbra, Portugal, March 2019), pp. 67–74.

[43] Lopes, R. R. F., Balaraju, P. H., Silva, A. T., Rettore, P. H., and Sevenich, P. Experiments with a queuing mechanism over ever-changing datarates in a VHF network. In *IEEE Military Communications Conference (MILCOM)* (Norfolk VA, USA, November 2019), pp. 712–717.

[44] Lopes, R. R. F., Loevenich, J., Sergeev, A., Rettore, P. H., and Sevenich, P. Quantifying the robustness to unplanned link disconnections in tactical networks. In *IEEE Conference on Military Communications (MILCOM)* (San Diego, USA, Nov. 2021). (under review).

[45] Lopes, R. R. F., Loevenich, J. F., Rettore, P. H., Eswarappa, S. M., and Sevenich, P. Quantizing radio link data rates to create ever-changing network conditions in tactical networks. *IEEE Access 8* (September 2020), 188015–188035.

[46] Lopes, R. R. F., Nieminen, M., Viidanoja, A., and Wolthusen, S. D. Reactive/proactive connectivity management in a tactical service-oriented infrastructure. In *International Conference on Military Communications and Information Systems (ICMCIS)* (Oulu, Finland, May 2017), pp. 1–8.

[47] Lopes, R. R. F., Rettore, P. H., Eswarappa, S. M., Loevenich, J., and Sevenich, P. Performance analysis of proactive neighbor discovery in a heterogeneous tactical network. In *IEEE International Conference on Military Communication and Information Systems (ICMCIS)* (2021), pp. 1–8.

[48] Lopes, R. R. F., Viidanoja, A., Lhotellier, M., Diefenbach, A., Jansen, N., and Ginzler, T. A queuing mechanism for delivering QoS-constrained web services in tactical networks. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2018).

[49] Lopes, R. R. F., Viidanoja, A., Lhotellier, M., Mazurkiewicz, M., Melis, G., Diefenbach, A., Ginzler, T., and Jansen, N. Trade-off analysis of a service-oriented and hierarchical queuing mechanism. In *IEEE 16th International Symposium on Network Computing and Applications (NCA)* (Oct 2017), pp. 1–4.

[50] Lund, K., Eggen, A., Hadzic, D., Hafsoe, T., and Johnsen, F. T. Using web services to realize service oriented architecture in military communication networks. *IEEE Communications Magazine 45*, 10 (October 2007), 47–53.

[51] Lund, K., Skjervold, E., Johnsen, F., Hafsøe, T., and Eggen, A. Robust web services in heterogeneous military networks. *IEEE Communications Magazine 48*, 10 (October 2010), 78–83.

[52] Małowidzki, M., Dalecki, T., Bereziński, P., Mazur, M., and Skarżyński, P. Adapting standard tactical applications for a military disruption-tolerant network. In *2016 International Conference on Military Communications and Information Systems (ICMCIS)* (2016), pp. 1–5.

[53] Małowidzki, M., Kaniewski, P., Matyszkiel, R., and Bereziński, P. Standard tactical services in a military disruption-tolerant network: Field tests. In *IEEE Military Communications Conference (MILCOM)* (Oct 2017), pp. 63–68.

[54] Merz, R., Schiöberg, H., and Sengul, C. Design of a configurable wireless network testbed with live traffic. In *Testbeds and Research Infrastructures. Development of Networks and Communities* (Berlin, Heidelberg, 2011), T. Magedanz, A. Gavras, N. H. Thanh, and J. S. Chase, Eds., Springer, pp. 189–198.

[55] Monge, G. *Mémoire sur la théorie des déblais et des remblais.* De l'Imprimerie Royale, 1781. `https://books.google.de/books?id=IG7CGwAACAAJ`.

[56] Morelli, A., Stefanelli, C., Tortonesi, M., Lenzi, R., and Suri, N. A proxy gateway solution to provide qos in tactical networks and disaster recovery scenarios. In *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet)* (New York, NY, USA, 2015), pp. 43–50.

[57] Nicholas, P. J., Tkacheff, J. C., and Kuhns, C. M. Analysis of throughput-constrained tactical wireless networks. In *IEEE Military Communications Conference (MILCOM)* (Nov 2014), pp. 916–921.

[58] Nosheen, I., Khan, S. A., and Khalique, F. A mathematical model for cross layer protocol optimizing performance of software-defined radios in tactical networks. *IEEE Access 7* (2019), 20520–20530.

[59] Nunen, J. A set of successive approximation methods for discounted markovian decision problems. *Zeitschrift für Operations Research 20* (1976), 203–208.

[60] Pease, S. G., Phillips, I. W., and Guan, L. Adaptive intelligent middleware architecture for mobile real-time communications. *IEEE Transactions on Mobile Computing 15*, 3 (March 2016), 572–585.

[61] Peyre, G., and Cuturi, M. Computational optimal transport. *Foundations and Trends in Machine Learning 11*, 5-6 (2019), 355–607.

[62] Poylisher, A., Sultan, F., Ghosh, A., Li, S.-w., Chiang, C. J., Chadha, R., Moeltner, K., and Jakubowski, K. QAM: A comprehensive QoS-aware middleware suite for tactical communications. In *IEEE Military Communications Conference (MILCOM)* (Nov 2011), pp. 1586–1591.

[63] Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, 2015.

[64] Puterman, M. L., and Shin, M. C. Modified policy iteration algorithms for discounted markov decision problems. *Management Science 24*, 11 (1978), 1127–1137. `https://EconPapers.repec.org/RePEc:inm:ormnsc:v:24:y:1978:i:11:p:1127-1137`.

[65] Ratliff, S., Jury, S., Satterwhite, D., Taylor, R., and Berry, B. Dynamic Link Exchange Protocol (DLEP). RFC 8175, RFC Editor, June 2017. `https://www.rfc-editor.org/rfc/rfc8175.txt`.

[66] Rettore, P. H., Loevenich, J., Lopes, R. R. F., and Sevenich, P. TNT: A tactical network test platform to evaluate military systems over ever-changing scenarios. *IEEE/ACM Trans. Netw.* (Mar. 2021). pre-print, doi:10.36227/techrxiv.14141501.

[67] Rettore, P. H., von Rechenberg, M., Loevenich, J., Rigolin F. Lopes, R., and Sevenich, P. A handover mechanism for Centralized/Decentralized networks over disruptive scenarios. In *MILCOM 2021 Track 2 - Networking Protocols and Performance (MILCOM 2021 Track 2)* (San Diego, USA, nov 2021). (under review).

[68] Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision 40* (2000), 2000.

[69] Ruffieux, S., Gisler, C., Wagen, J., Buntschu, F., and Bovet, G. TAKE - tactical ad-hoc network emulation. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2018), pp. 1–8.

[70] Schmitzer, B. Stabilized sparse scaling algorithms for entropy regularized transport problems, 2019.

[71] Scott, K., Refaei, T., Trivedi, N., Trinh, J., and Macker, J. P. Robust communications for disconnected, intermittent, low-bandwidth (DIL) environments. In *IEEE Military Communications Conference (MILCOM)* (Nov 2011), pp. 1009–1014.

[72] Shannon, C. E. A mathematical theory of communication. *Bell Syst. Tech. J. 27*, 3 (Jul. 1948), 379–423.

[73] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature 529* (2016), 484–503. `http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html`.

[74] Strogatz, S. H. *Infinite Powers.* Houghton Mifflin Harcourt, Boston, Massachusetts, 2019.

[75] Suri, N., Breedy, M. R., Marcus, K. M., Fronteddu, R., Cramer, E., Morelli, A., Campioni, L., Provosty, M., Enders, C., Tortonesi, M., and Nilsson, J. Experimental evaluation of group communications protocols for data dissemination at the tactical edge. In *2019 International Conference on Military Communications and Information Systems (ICMCIS)* (May 2019), pp. 1–8.

[76] Suri, N., Hansson, A., Nilsson, J., Lubkowski, P., Marcus, K., Hauge, M., Lee, K., Buchin, B., Misirhoglu, L., and Peuhkuri, M. A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2016), pp. 1–8.

[77] Suri, N., Nilsson, J., Hansson, A., Sterner, U., Marcus, K., Misirlioğlu, L., Hauge, M., Peuhkuri, M., Buchin, B., in't Velt, R., and Breedy, M. The angloval tactical military scenario and experimentation environment. In *International Conference on Military Communications and Information Systems (ICMCIS)* (May 2018), pp. 1–8.

[78] Sutton, R. S., and Barto, A. G. *Reinforcement learning: An introduction.* MIT press, 2018.

[79] T. Clausen, C. Dearlove, P. J., and Herberg, U. RFC 7181: The optimized link state routing protocol version 2, Apr. 2014. `https://tools.ietf.org/html/rfc7181`.

[80] Tang, N., and Lin, Y. Fast encoding and decoding algorithms for arbitrary (n,k) Reed-Solomon codes over F2m. *IEEE Communications Letters 24*, 4 (2020), 716–719.

[81] Wigner, E. P. *The Unreasonable Effectiveness of Mathematics in the Natural Sciences.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1995, pp. 534–549. `https://doi.org/10.1007/978-3-642-78374-6_41`.

[82] Zeng, H., Kwak, K. J., Deng, J., Fu, B., Xiao, Y., and Jeski, J. Proactive and adaptive reconfiguration for reliable communication in tactical networks. In *Defense Transformation and Net-Centric Systems 2012* (2012),

R. Suresh, Ed., vol. 8405, International Society for Optics and Photonics, SPIE, pp. 115 – 123. https://doi.org/10.1117/12.919436.

List of Figures

List of Tables

Acronyms

C2 Command and Control. 1, 6, 16

C3 Consultation, Command and Control. 5, 6, 9

CDF Cumulative Distribution Function. 48, 49, 51–53, 69, 70

DDP Discrete Dynamic Programming. 47, 50

DKW Dvoretzky-Kiefer-Wolfowitz. 48, 49

DLEP Dynamic Link Exchange Protocol. 7

EDF Empirical Distribution Function. 48, 49, 52, 66, 68

HF High Frequency. 16, 56

i.i.d. independent and identically distributed. 48

ICMP Internet Control Message Protocol. 66

IP Internet Protocol. 3, 4, 15, 18–20, 62, 67

K-S test Kolmogorov-Smirnov Test. 48, 53, 68

MANET Mobile Adhoc Network. 9, 10

MC Markov Chain. vi, 2, 3, 8, 12, 20, 21, 23–25, 27–29, 31, 33, 36, 45, 46, 49, 50, 56, 58, 59

MDP Markov Decision Process. vi, x, 2, 4, 10, 17, 19, 20, 35, 45, 47, 49–51, 66, 67, 75, 77, 87

NATO North Atlantic Treaty Organization. 5, 6

OLSR Optimized Link State Routing. 9, 62

OT Optimal Transport. vi, x, 14, 35–38, 41, 42, 44

QoS Quality of Service. 6, 15

RL Reinforcement Learning. 9, 10

SatCom Satellite Communications. 16

SDR Software Defined Radio. 9

SNMP Simple Network Management Protocol. 7, 62

SOA Service Oriented Architecture. 7

SRR Smart Robust Routing. 10

TCP Transmission Control Protocol. 9

TN Tactical Network. vi, ix, 1–3, 5–10, 12, 15–17, 19, 45, 49, 54, 75

TNT Tactical Network Test. 8

TS Tactical Communication System. vi, ix, 1–10, 14–20, 23, 34, 54, 58, 70, 75, 77

TtR Time to Resume IP Data Flows. vi, 2–4, 17, 35, 48, 49, 51, 52, 61, 66–70, 73, 75, 87, 88

UDP User Datagram Protocol. 9

UHF Ultra High Frequency. 16

VHF Very High Frequency. 16, 56, 61, 62, 71