TECHNISCHE
UNIVERSITÄT
DARMSTADT

CONNECTING VEHICLES TO THE INTERNET

Strategic Data Transmission for Mobile Nodes using Heterogeneous Wireless Networks

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

TOBIAS RÜCKELT, M.SC.

Geboren am 18. April 1987 in Fulda

Vorsitz:	Prof. Dr.-Ing. Franko Küppers
Referent:	Prof. Dr.-Ing. Ralf Steinmetz
Korreferent:	Prof. Dr. Ioannis Stavrakakis

Tag der Einreichung:	20. Juni 2017
Tag der Disputation:	5. Oktober 2017

Hochschulkennziffer D17
Darmstadt 2017

Tobias Rückelt, M.Sc., *Connecting Vehicles to the Internet, Strategic Data Transmission for Mobile Nodes using Heterogeneous Wireless Networks* © 2017

ABSTRACT

With the advent of autonomous driving, the driving experience for users of connected vehicles changes, as they may enjoy their travel time with entertainment, or work productively. In our modern society, both require a stable Internet access. However, future mobile networks are not expected to be able to satisfy application Quality of Service (QoS) requirements as needed, e.g. during rush hours. To address this problem, this dissertation investigates data transmission strategies that exploit the potential of using a heterogeneous wireless network environment. To this end, we combine two so far distinct concepts, firstly, network selection and, secondly, transmission time selection, creating a joint time-network selection strategy. It allows a vehicle to plan delay-tolerant data transmissions ahead, favoring transmission opportunities with the best prospective flow-network matches.

In this context, our first contribution is a novel rating model for *perceived* transmission quality, which assesses transmission opportunities with respect to application QoS requirement violations, traded off by monetary cost. To enable unified assessment of all data transmissions, it generalizes existing specialized rating models from network selection and transmission time selection and extends them with a novel throughput requirement model.

Based on that, we develop a novel joint time-network selection strategy, Joint Transmission Planning (JTP), as our second contribution, planning optimized data transmissions within a defined time horizon. We compare its transmission quality to that of three predominant state-of-the-art transmission strategies, revealing that JTP outperforms the others significantly by up to 26%. Due to extensive scenario variation, we discover broad stability of JTP reaching 87-91% of the optimum.

As JTP is a planning approach relying on prediction data, the transmission quality is strongly impaired when executing its plans under environmental changes. To mitigate this impact, we develop a transmission plan adaptation as our third contribution, modifying the planned current transmission online in order to comply with the changes. Even under strong changes of the vehicle movement and the network environment, it sustains 57%, respectively 36%, of the performance gain from planning.

Finally, we present our protocol Mobility management for Vehicular Networking (MoVeNet), pooling available network resources of the environment to enable flexible packet dispatching without breaking connections. Its distributed architecture provides broad scalability and robustness against node failures. It complements control mechanisms that allow a demand-based and connection-specific trade-off between overhead and latency. Less than 9 ms additional round trip time in our tests, instant handover and 0 to 4 bytes per-packet overhead prove its efficiency.

Employing the presented strategies and mechanisms jointly, users of connected vehicles and other mobile devices can significantly profit from the demonstrated improvements in application QoS satisfaction and reduced monetary cost.

KURZFASSUNG

Autonomes Fahren befreit Fahrzeugnutzer in Zukunft von ihrer Fahraufgabe, sodass sie ihre Reisezeit entspannt bei Entertainment genießen oder produktiv nutzen können. In unserer modernen Gesellschaft erfordert beides eine stabile Internetverbindung. Die vorliegende Dissertationsschrift addressiert diese Herausforderung durch innovative Datenübertragungsstrategien mit kombinierter Nutzung heterogener Netzwerke. Dazu werden zwei in der aktuellen Forschung verfolgte Konzepte, die Netzwerk-Auswahl und die Auswahl des Übertragungszeitpunktes, zu einer gemeinsamen Zeit-Netzwerk-Auswahl Strategie kombiniert. Damit werden für jede Übertragung die am besten geeigneten Netzwerke in einem zeitlichen Planungshorizont gewählt.

In diesem Kontext ist der erste Kernbeitrag eine Funktion zur Bewertung von der *wahrgenommenen* Qualität von Übertragungsmöglichkeiten, die erwartete Verletzungen applikationsspezifischer Anforderungen an die Kommunikation gegenüber entstehenden Übertragungskosten abwägt. Diese Bewertungsfunktion verallgemeintert spezialisierte Modelle aus den Bereichen der Netzwerk-Auswahl und der Auswahl des Übertragungszeitpunktes und erweitert sie um ein Datendurchsatzmodell, welches eine einheitliche Bewertung von Datenflüssen ermöglicht.

Darauf basierend stellt eine Strategie zur Zeit-Netzwerk-Auswahl, genannt Joint Transmission Planning (JTP), den zweiten Kernbeitrag dar, welche Datenübertragungen optimiert für einen gewissen Zeithorizont plant. Ein Vergleich mit führenden Stragegien aus verwandten Arbeiten zeigt für JTP signifikante Leistungsgewinne von bis zu 26%. Bei Szenariovariationen erreicht JTP als einziger Ansatz konstant hohe Leistungen von 87-91% im Vergleich zum Optimalwert.

Da JTP ein Planungsverfahren ist und auf Vorhersagen operiert, nimmt die Übertragungsqualität für eine Planausführung bei verändertem Umfeld drastisch ab. Der dritte Kernbeitrag addressiert dieses Problem mit einem Verfahren zur Adaption der aktuellen Übertragung während der Planausführung. Selbst bei starken Änderungen gegenüber der Bewegungs- und Netzwerkvorhersage bewahrt das Adaptionsverfahren 57% bzw. 36% des Leistungsgewinns durch JTP.

Der vierte Kernbeitrag umfasst ein neues Mobilitäts-Management-Konzept, genannt MoVeNet, welches verfügbare Netzwerkressourcen zur flexiblen Nutzung bereitstellt, ohne aktive Verbindungen zu unterbrechen. Seine verteile Architektur schafft eine hohe Skalierbarkeit und Robustheit gegenüber Ausfällen von Knoten. Sich ergänzende Mechanismen bieten für Verbindungen nach Bedarf eine hohe Übertragungseffizienz oder eine niedrige Latenz. Verzögerungen im Routing von weniger als 9 ms in Tests, augenblickliche Netzwerkwechsel und nur 0 bis 4 Bytes Overhead pro Paket zeigen, dass MoVeNet einen effizienten Lösungsansatz zur Ausführung von Übertragungsplänen darstellt.

Somit kann die wahrgenommene Qualität der Internetverbindung von Fahrzeugen sowie anderen mobilen Knoten durch stratigische Datenübertragung in heterogenen Netzen signifikant verbessert werden.

CONTENTS

1 INTRODUCTION 1
- 1.1 Motivation 1
- 1.2 Goal and Research Questions 2
- 1.3 Contributions 4
- 1.4 Thesis Structure 5

2 FUNDAMENTALS 7
- 2.1 Connected Vehicles 7
- 2.2 Network Environment 8
  - 2.2.1 Mobile Cellular Networks 8
  - 2.2.2 Dedicated Short Range Communication 9
  - 2.2.3 Consumer WiFi 9
  - 2.2.4 Short-Range Network Connectivity Duration 10
- 2.3 Connectivity Prediction 10
- 2.4 Constrained Optimization 11

3 RELATED WORK 13
- 3.1 Network Selection 13
  - 3.1.1 Network-Controlled and Network-Assisted Network Selection 14
  - 3.1.2 Client-Controlled Network Selection 16
  - 3.1.3 Conclusions 17
- 3.2 Transmission Time Selection 18
  - 3.2.1 Relevant Approaches 18
  - 3.2.2 Conclusions 20
- 3.3 Mobility Management 21
  - 3.3.1 Problem Statement and General Concept 21
  - 3.3.2 Direct and Proxy-Based Approaches 22
  - 3.3.3 Conclusions 26

4 STRATEGIC DATA TRANSMISSION 29
- 4.1 Transmission Rating Model 30
  - 4.1.1 Model Components 31
  - 4.1.2 Formal Transmission Rating Model Definition 32
- 4.2 Joint Time-Network Selection in Transmission Planning 42
  - 4.2.1 Transmission Planners with Different Time-Impact 43
  - 4.2.2 Heuristics 47
- 4.3 Evaluation 51
  - 4.3.1 Evaluation Metrics and Dependent Variables 51
  - 4.3.2 Evaluation Setup and Independent Variables 54
  - 4.3.3 Impact of Time Selection: Variation of the Planning Horizon Length 56
  - 4.3.4 Impact of the Number of Networks 61
  - 4.3.5 Impact of the Data Traffic Load 64
  - 4.3.6 Impact of the Number of Flows 68

4.3.7 Impact of the Monetary Cost Weight 70
4.4 Summary and Conclusions 73
5 TRANSMISSION PLAN ADAPTATION 75
5.1 Prediction Error Models 75
5.1.1 Network Characteristics Prediction Error 76
5.1.2 Movement Prediction Error 78
5.1.3 Data Flow Prediction Error 80
5.1.4 Prediction Error Examples 80
5.1.5 Prediction Error Adjustment 82
5.2 Transmission Plan Adaptation Algorithm 83
5.2.1 Adaptation Concept 84
5.2.2 Plan Execution Algorithm 84
5.2.3 Extended Data Release Mechanism 86
5.2.4 Location Reference Mechanism 87
5.2.5 Flow Prediction Error Handling 88
5.2.6 Adaptation Design Discussion 90
5.3 Evaluation 92
5.3.1 Evaluation Setup 92
5.3.2 Impact of Network Prediction Errors 92
5.3.3 Impact of Movement Prediction Errors 97
5.3.4 Impact of Data Flow Prediction Errors 100
5.3.5 Impact of Combined Prediction Errors 104
5.3.6 Execution Time Analysis and Re-Planning 108
5.4 Summary and Conclusions 108
6 MOBILITY MANAGEMENT FOR TRANSMISSION PLAN EXECUTION 111
6.1 System Requirements 112
6.2 MoVeNet Architecture 113
6.2.1 Mobile Agent – The In-Vehicle Entity 114
6.2.2 Data Agent – Light-Weight Proxy 114
6.2.3 Control Agent – System Orchestration 115
6.2.4 Routing and Signaling Concept 115
6.3 MoVeNet Protocol Details 116
6.3.1 IP Address Mapping System 116
6.3.2 Latency Optimization Considerations 121
6.3.3 Initialization Process 122
6.3.4 IP Table Synchronization 125
6.3.5 Identification Header 129
6.3.6 Event-Based Retransmission Trigger 130
6.3.7 Security Considerations 131
6.3.8 Robustness Considerations 132
6.3.9 Further Overhead Reduction 132
6.4 Analytical Evaluation 133
6.4.1 Design Implications 133
6.4.2 Handover Delay Analysis 133
6.5 Simulative Evaluation 135
6.5.1 Simulation Setup 135

6.5.2 Handover Delay 137
6.5.3 Retransmission Trigger 137
6.6 Prototypical Evaluation 138
6.6.1 Linux Prototype Design 138
6.6.2 Prototype Evaluation Setup 140
6.6.3 Connection Setup and Round Trip Time 140
6.6.4 Throughput Impact and Scalability 143
6.7 Transmission Planning as Control Entity for MoVeNet 143
6.8 Summary and Conclusions 145
7 CONCLUSIONS AND OUTLOOK 147
7.1 Conclusions 147
7.2 Outlook 150

BIBLIOGRAPHY 151
A APPENDIX 169
A.1 Complexity Analysis of the Transmission Planning Problem 169
A.2 Random Transmission Planning Algorithm 169
A.3 Data Flow Types 170
A.3.1 Default Data Flow Template 171
A.3.2 Interactive Data Flow 171
A.3.3 Conversational Data Flow 171
A.3.4 Bufferable Data Flow 171
A.3.5 Background Data Flow 172
A.4 Parameter Optimization of Joint Transmission Planning 174
A.5 Further Transmission Plan Evaluation Results 176
A.5.1 Number of Networks 176
A.5.2 Number of Data Flows 178
A.5.3 Monetary Cost Weight 179
A.5.4 Adaptation Results with Relative Optimization Potential 180
A.6 Example of Applied Prediction Error Model with SMAPE 0.5 181
B PUBLICATIONS 183
B.1 Main Publications 183
B.2 Co-Authored Publications 184
C SUPERVISED STUDENT WORKS 185
C.1 Master Theses 185
C.2 Bachelor Theses 185
C.3 Others 185
D DANKE 187
E CURRICULUM VITAE 189
F ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG 191

1 INTRODUCTION

> *In terms of overall data volumes, connected cars don't present much of a problem. But network resource management is not based on total traffic volume.*
>
> — *Matt Hatton, 2015*

With the current advent of connected vehicles, new in-vehicle services and applications get into focus, which benefit from external data sources. Hence, the wireless Internet connection gains importance for the automotive industry [58, 140, 148]. In the following Section 1.1, we present current technology trends that will change our experience of mobility, motivating our research on automotive Internet connectivity. This change raises new questions in the topic of Internet connectivity management, which we present together with the goal of this dissertation in Section 1.2. Finally, we highlight our contributions in Section 1.3 and close with the structure of this thesis in Section 1.4.

1.1 MOTIVATION

Connected vehicle data services support driver assistance systems, which rapidly converge from driving task automation towards autonomous driving. Finally, autonomous driving leads to a complete change in driver experience. The driver is no longer an essential part of the vehicle control loop and can focus on other tasks. This change offers new opportunities, using the vehicle as *mobile office*, exploiting the travel time to complete work tasks, or as *mobile living room*, relaxing with entertainment [92]. In our modern society, both require a stable Internet connection.

Highly automated driving is still in the early stage of development. Therefore, the driver still has to pay full attention to the driving task. This fact limits the type of existing connected vehicle services: they focus on functions that either support the driver with additional information for his driving task or that require his attention only occasionally during a trip. This covers, e.g., providing online traffic information or streaming music. Other services provide vehicle data to the user's smartphone or let him remotely control non-driving related functions, like setting up a route to the navigation system, starting the heater or locking the car [75, 211, 47, 24]. More advanced services provide the user with crowd-sourced location information or controller firmware updates [205]. As these services have rather low requirements to the Internet connection, consequences on the perceived system performance through, e.g., a high connection latency of a few seconds or a temporary connection loss, are negligible. Furthermore, bandwidth requirements for this information or control request services are very low. Hence, the connectivity can easily be achieved using a conventional mobile network connection.

However, this fact changes as soon as the driver is out of the vehicle control loop and can spend the majority of his travel time productively or relaxing. This is already the case for passengers in a bus or train [75, 52]. Hence, WiFi for their personal devices is often already provided. However, permanent access to online resources during work requires a responsive Internet connection. Multimedia streams in steadily improving quality call for high data rates. Furthermore, voice and video telephony converge towards all IP transmission [222] and impose tough requirements on latency, jitter and network continuity [199, 201]. Especially during rush hours, satisfying the future communication demand from connected cars is considered as a critical challenge for network operators [139]. Conclusively, different kinds of connected car applications selectively impose tough requirements to the wireless access. A single mobile network, even with expected future resources, is not able to satisfy these requirements pervasively during a trip [139]. How can we solve this conflict, providing passengers the future travel experience they desire? To address this question, we present the goals of this dissertation in the following section and clarify the dedicated research questions arising from this scenario.

1.2 GOAL AND RESEARCH QUESTIONS

The goal of this thesis is the optimization of the user's *perceived* transmission quality in connected vehicles through intelligent network management to satisfy the requirements of each active communication. However, instead of upgrading the bare transmission technologies, we develop client-based strategies that use available network resources of multiple operators smartly. Hence, we investigate and design strategies to mitigate the effects of network resource shortages by distributing data transmission reasonably over multiple networks and time slots, introducing an explicit *time-network selection*. With this goal and based on insights obtained in our analysis of related work, we formulate three research questions in the following and detail the corresponding research gaps.

1. How can the *perceived* quality of all data transmissions of a client be rated?

   The perceived transmission quality refers to a subset of Quality of Experience and covers only transmission-related parameters, excluding factors from the device and context [173]. Rating functions assess the quality and reflect optimization objectives. Hence, inappropriate or incomplete rating functions applied for mechanism design may render solutions ineffective. Existing models for mobile data transmission quality rating lack general applicability, each focusing on special data traffic types, i.e. video transmissions, and are unable to assess the quality for other types of data transmissions [13, 132]. Even if the majority of data traffic is covered, most transmission rating approaches do not consider a satisfactory rating [5], defined as one for which a surplus of network characteristics makes no difference as long as application QoS requirements are satisfied. Hence, they do not target the *perceived* Internet access quality, but an absolute one. Moreover, to the best of our knowledge, there exists no approach that sufficiently covers both dimensions for data transmission distribution: time and networks. This renders existing transmission rating models ineffective for general application. Hence, there is a gap for a general satisfactory transmission performance rating model for data distribution over networks and time.

2. How can data be transmitted strategically in order to improve the perceived transmission quality?

   Approaches in related work address strategic data transmission in two distinct ways, distributing transmissions (1) over networks or (2) over time. Moreover, these approaches usually pursue distinct goals. On the one hand, network selection strategies focus on the current situation of the mobile node. Their goal is to select the best-possible network for current data transmissions [212, 16], also dealing with incomplete information about the environment [217].

   On the other hand, there exist delayed-offloading strategies, distributing data transmission over time. They pursue the goal to move as much data traffic as possible from cellular to WiFi networks [145, 131]. Therefore, they focus on delay-tolerant data and seek to transmit it at a point in time, when WiFi is available. However, they ignore the flow-network matching, which, in sharp contrast, is the primary focus of network selection.

   Even though the authors of these approaches never state it, they implicitly follow a common goal: Avoiding the impact of network overload during resource bottlenecks through the reasonable distribution of data transmissions over a certain dimension. Related works show that transmission distribution in each dimension, network and time, yields significant performance benefits. From these findings, we identify a research gap for combined data transmission strategies, investigating the performance of joint time-network selection.

3. How can cross-operator wireless network resources be accessed flexibly and efficiently for transmission strategy execution?

   Flexible routing of data is fundamental for strategic data transmission. It is addressed by multi-homing and mobility management protocols, which decouple packet routing via networks from connection identification in the transport layer, hiding route changes to avoid transmission interruptions. Even though there exists a huge variety of those protocols, there is no efficient approach satisfying the requirements of the connected vehicle scenario.

   For strategic data transmission, each non-covered transmission acts as a disturbance to the strategy, diminishing its benefits. Thus, all data should be covered, which disqualifies transport layer approaches. Furthermore, many approaches focus on intra-operator mobility management only, implementing functions at internal network entities of the operator. Since operators do not allow external control of their own hardware, e.g. access points or software defined network routers, these approaches are infeasible for implementing cross-operator transmission strategies. Without relying on the network operators, many remaining approaches establish management contexts with their communication partners, which have to implement the specific protocol, i.e. requiring changes in the operating systems. This limits possible communication to communication partners that implement the protocol. Remaining protocols, like Mobile IPv6, are proven to be inefficient and lack robustness. Accordingly, we identify the research gap of a cross-operator mobility management protocol that satisfies requirements of the connected vehicle scenario and enables strategic data transmission via heterogeneous wireless networks.

1.3 CONTRIBUTIONS

This dissertation treats data transmission optimization strategies for the vehicular Internet access. The main contribution of this work is the generalization and combination of the two distinct concepts of transmission time selection and network selection into a joint time-network selection. We show that combining the two concepts offers a significantly higher optimization potential for data transmission than separate optimization. Moreover, we present our algorithms for strategic transmission planning and reaction to environmental changes. To execute these transmission plans, we additionally develop a mobility management architecture, which constructs the bridge from theory towards practical systems. The following four contributions define the keystones of this dissertation in order to reach our major goal: improving the perceived transmission quality of vehicle occupants.

1. Our first contribution is the generalization of existing transmission rating models in order to enable unified assessment of data transmissions, addressing the perceived transmission quality according to our first research question. Our novel transmission rating model combines two main objectives: Firstly, application QoS requirement *satisfaction* and, secondly, monetary cost to trade-off economic aspects. To this end, our model integrates components of models from network selection and from models of transmission time selection. Moreover, we extend it with a novel throughput requirement model, which generalizes employed models from both areas for unified assessment of all data transmissions. Thus, it generalizes existing rating models in order to treat transmissions in a unified way and assesses the user's *perceived* transmission quality.

2. As our second contribution, we design the explicit time-network selection strategy Joint Transmission Planning (JTP). The approach creates transmission plans, in which for each transmission the best-matching transmission opportunity within a certain planning time horizon is selected, while trading off monetary cost, addressing our second research question. JTP significantly outperforms the transmissions of leading state-of-the-art strategies by 7-26%, reaching even under parameter variation robustly 87-91% of the scenario's optimization potential.

3. The third contribution is our transmission plan adaptation algorithm, which addresses transmission robustness against environmental changes. JTP assumes perfect knowledge about future available data flows and networks. We show that simple execution of JTP's transmission plans lacks resilience against prediction errors that occur due to incomplete information or environmental changes. To react dynamically on observed environmental changes, we design an adaptation algorithm that modifies the current time slot of an existing transmission

plan. It consists of three heuristics that preserve beneficial temporal transmission patterns and network selection from the plan and, at the same time, enable a dynamic reaction to environmental changes. The result sustains a significant share of the performance gain achieved from JTP for moderate prediction errors, in particular of the movement (61%), of the networks (66%) and of the data flows (56%). For large prediction errors, its performance converges towards the performance of the underlying opportunistic transmission approach, providing a robust lower bound.

4. As our final contribution, we push transmission planning towards real systems. Thus, we present a network architecture and protocol that we call MoVeNet: Mobility Management for Vehicular Networking, addressing the third research question. MoVeNet pools currently available network resources in order to distribute data transmissions over the available wireless cross-operator networks. Its distributed architecture ensures system robustness against node failures and a low additional round trip time of less than 9 ms in our tests. Complementary control mechanisms provide a connection-specific trade-off between low latency or high efficiency. Furthermore, a novel IP mapping approach reduces per-packet overhead to only 0 to 4 bytes though address multiplexing, while a new retransmission trigger optimizes TCP performance by up to 64% additionally transmitted packets in our tests in environments with sparse network availability. We propose a concept to integrate the long-term transmission planning as well as the adaptation algorithm into the MoVeNet architecture to provide a full system design that facilitates strategic data transmission for mobile nodes using heterogeneous wireless networks.

The proposed design improves the perceived transmission quality for mobile nodes through the strategic exploitation of network resources, addressing expected future resource bottlenecks of mobile networks without their modification.

1.4 THESIS STRUCTURE

In the next two chapters, we provide selected fundamentals to simplify understanding of the scenario characteristics and the developed time-network selection and discuss relevant related work, detailing the scientific research gaps. In Chapter 4, we present our novel transmission rating model and the design of our heuristic approach Joint Transmission Planning (JTP) for explicit time-network selection. We evaluate its performance and compare it to existing concepts. To investigate the robustness of JTP against prediction errors, i.e. environmental changes, we create and apply prediction error models for the vehicle scenario in Chapter 5. Subsequently, we present the design of our adaptation approach, which ensures resilience against identified kinds of prediction errors. Finally, in Chapter 6, we present our designed protocol and architecture MoVeNet: Mobility Management for Vehicular Networking. It follows a distributed design and drives strategic transmission planning towards real systems. Thus, it composes all aspects of the presented approach in a single system design, targeting improvement of the vehicle user's perceived transmission quality.

2 FUNDAMENTALS

This dissertation focuses on the optimization of the vehicle Internet connection using heterogeneous wireless networks. To give a background about the environment under optimization, we present an overview about current and approaching technologies, systems and services of connected vehicles as well as network technologies. In the last section of this chapter, we provide selected basics on optimization to simplify understanding of Chapter 4 on transmission planning and Chapter 5 on transmission plan adaptation.

2.1 CONNECTED VEHICLES

Vehicles services profit from external information beyond in-vehicle sensing capabilities. An early adopter of this idea is the FM Radio Data System (RDS) based Traffic Message Channel (TMC) [61], initially designed in 1986 and deployment in Germany in 2005. It employs a 60 bits/s broadcast channel via radio stations to distribute up-to-date traffic information, used to enhance vehicle navigation. With the rise of bidirectional mobile communication, the vision of connected vehicles emerged early, for example, with the project SOCRATES (1991) [37] demonstrating basic information services using cellular radio. Faster data transmissions with 2G and 3G networks increased the speed of development and projects as CVIS (2007) [120], introducing Traffic Telematic services to support the driver with personalized up-to-date information.

While connected vehicle research projects focused mainly on the safety and traffic efficiency, series development pushed personalized connectivity services as unique selling points into vehicles.

Past and ongoing research projects on connected vehicles, e.g. SimTD (2009) [193], DriveC2X (2011) [54], CONVERGE [45] (2013) and IMAGinE (2016), focus mainly on safety, e.g. hazard warnings as electronic brake light, and road efficiency services [180, 36], e.g. green wave assistance or crowd-sourced map data [31]. They focus on the ad-hoc technology 802.11p [96] or hybrid communication together with cellular networks. In contrast, the development in series pushed systems towards implementing services as unique selling points, integrated into the infotainment system [75, 211, 47, 24, 205]. Typical examples are dynamic navigation incorporating up-to-date information, music streaming or firmware updates. These services either support the driver, simplify vehicle maintenance or improve the comfort and convenience. Unlike the services of systems in the presented research projects, they do not rely on other vehicles and show effects on demand, i.e. can be triggered actively from the user. Since most of these services are not time-critical and focus on providing limited information, the requirements for the Internet connection for today's systems are quite low and can usually be satisfied by existing mobile networks.

Considering, in particular, future highly automated driving functions, we expect an essential extension of the deployed service types by always active information update processes, enriching the sensed environment with external data [102, 104] in order to improve automated driving reliability and efficiency. As the vehicle occupants do not have to focus on the driving task anymore and can spend their time as desired [140], this service extension will be accompanied by a substantial increase of common Internet service usage [58], raising the motivation of this dissertation.

2.2 NETWORK ENVIRONMENT

In this section, we present different wireless Internet access technologies that connected vehicles may use and highlight their different characteristics to clarify the advantages and disadvantages of their use in heterogeneous Internet access.

2.2.1 *Mobile Cellular Networks*

Broad coverage and varying QoS characterizes mobile cellular networks, depending on available technology at the current location and network load [166]. In particular, varying QoS characteristics originate from their grown structure. Mobile network operators built up those networks over the past three decades, improving them gradually with new technologies to provide mainly higher data rates. While today 2G networks are nearly ubiquitous with a throughput of only a few hundred kbits/s for data transmissions and high latency of 300-1000ms [70], the targets for 3G, 4G and 5G are usually providing 10 times higher throughput and half the latency compared to the predecessor technology. Accordingly, for current LTE networks targeting 4G, the throughput reaches in practical systems usually about 50-600 Mbits/s [20] while latency is about 30-70ms [106]. As the goal of mobile network operators is the maximization of refunds from their investments, they tend to improve network quality preferably in areas with a high population density. This way, most users can profit from those investments at the same time. Since this strategy is favorable for both, network operators and the customers, the strategy is reasonable. However, there is a backside of the medal for the connected vehicle scenario. Street networks interconnect areas with high population densities. Therefore, they lead mainly through areas of low population density. Focusing on the motivational scenario, future occupants of highly automated vehicles will especially demand Internet access for mobile office or entertainment during their time on motorways because those sections provide the most comfortable travel characteristics with low acceleration.

The demand for Internet access in the connected vehicle scenario contrasts with today's mobile network coverage.

According to this, demand for Internet access in the connected vehicle scenario contrasts with today's mobile network coverage. Limited high data speed network resources are provided in many regions using the LTE 800MHz band, delivering throughput of 50MBits/s shared from all clients for a covered area with a range up to 10km. Furthermore, the monetary cost for transmission has to be considered as an important factor for network users, since the data amount of high speed data traffic is usually contractually limited.

2.2.2 *Dedicated Short Range Communication*

In contrast to mobile networks, Dedicated Short Range Communication (DSRC) [96] is a WiFi-based vehicular ad-hoc technology with the goal to exchange information between vehicles as well as with the infrastructure. DSRC is allocated in the 5.9 GHz spectrum, covering seven dedicated channels in Europe, each with 10MHz bandwidth for 6-12 Mbits/s data rates each. Furthermore, the standard proposes to use the close-by unlicensed WiFi spectrum in the 5.8GHz range. According to the European ETSI ITS-G5 standard [68], the dedicated channels are regulated in their use as summarized in the following listing.

- ITS road safety (G5A): one channel for control and basic safety, two channels for further safety services
- ITS non-safety applications (G5B): two channels, in particular for road traffic efficiency enhancement
- Future ITS applications (G5D): reserved for future use

This WiFi-based technology is optimized for high robustness, low latency of less than 30ms and longer ranges to the expense of lower data rates. It transmits messages usually in broadcast mode. Furthermore, Road Side Units (RSU) serve for exchanging information with service providers and may especially be located at road junctions and highly frequented areas. Different standards propose protocols for IP packet transmission via 802.11p networks. The European version is known as GN6 standard [67], transporting IPv6 packets via the GeoNetworking protocol [66] with the option to use multi-hop communications. In contrast, a competing architecture Communication Access for Land Mobiles (CALM) [101] for 802.11p provides a low-overhead transport protocol implementing single hop communication, designed to support unicast IP transfer. Hence, we can assume that 802.11p may be used to transport IP packets in the future.

The planned deployment pattern of RSUs contrasts from this of mobile networks. Instead of improving quality in areas of high population density, RSU networks are especially built up to support vehicular traffic safety and efficiency and will therefore be placed in areas of high traffic density, including motorways.

We can conclude that DSRC and mobile networks will complement each other in spatial coverage as well as communication characteristics, providing Internet access with lower latencies but lower throughput for potentially lower monetary cost.

2.2.3 *Consumer WiFi*

Consumer WiFi is designed for considerable high data rates within an unmanaged channel. Different standards share two spectra in the 2.4 GHz and 5.8 GHz region. Robustness and range are subordinate design goals. Transmission is unicast-based between the station and a mobile client, which allows the two entities to negotiate on data rates for higher robustness to optimize for transmission speeds in different situations.

A seasoned standard is 802.11g, which can reach gross data rates of up to 54 MBit/s in the 2.4GHz frequency region and is already widely replaced by the faster

802.11n [95], which can couple two channels and reach theoretical data rates of up to 150 MBit/s. Even though it is usually not implemented in most devices, MIMO techniques can even quadruple this speed. New devices integrate transceivers of the 802.11ac standard [97], using the 5.8 GHz band. It is can couple up to 8 channels to reach a data rate of up to 867 Mbit/s. Usually, devices implement 3x3 MIMO, arriving at a gross speed of up to 1300 Mbit/s. Furthermore, beam-forming is a major subject of the 802.11ac specification, providing high data rates even at higher ranges, which is favorable for the connected vehicle scenario.

Beam-forming is addressed in specifications, providing high performance even at higher ranges. This is favorable for connected vehicles.

An upcoming WiFi standard is 802.11ah [98], which – in contrast to the previously named – uses a 900 MHz spectrum. It is designed for the Internet of Things and therefore focuses on supporting high device numbers and low energy consumption. A similar approach is followed from the LoRa-Alliance [136]. They specified a WiFi technology in a similar band with similar goals. They might be used to transmit small amounts of delay-tolerant data via wide ranges.

2.2.4 *Short-Range Network Connectivity Duration*

In Table 1, we present typical connectivity durations in seconds for passing short-range networks, e.g. WiFi and DSRC, with a given speed [184], derived from passing the diameter of a circular covered area. Selected ranges are based on short-range network measurements of Gozalvez [81]. Considering a default highway scenario with a vehicle speed of 130 km/h and an access point (roadside station) range of 400 meters, we receive a theoretical duration of 17.09 seconds for a connection of a passing vehicle.

High vehicle speeds lead to short connectivity durations to short-range networks.

Table 1: Connection duration in seconds of a vehicle passing a short-range access point

AP range	**50 m**	**100 m**	**200 m**	**400 m**	**800 m**	**1500 m**
50 km/h	5.56	11.11	22.22	44.44	88.89	166.67
80 km/h	3.42	6.94	13.89	27.78	55.56	104.17
100 km/h	2.78	5.56	11.11	22.22	44.44	83.33
130 km/h	2.14	4.27	8.55	**17.09**	34.19	64.10
180 km/h	1.55	3.09	6.17	12.35	24.69	46.30

2.3 CONNECTIVITY PREDICTION

Connectivity planning methods, as investigated in this dissertation, require a prediction of future available network resources, relying on node mobility. Hence, both have to be predicted. For mobility prediction of vehicles, we assume routes to be planned and known, especially in the scenario of future autonomous vehicles. In the case of an unknown destination, algorithms can estimate a most-probable-path [33, 34]. Assuming that vehicles tend to follow main roads and head towards a persistent direction, or re-use same routes over time, these algorithms estimate correctly in most cases. Using crowd-sensed probe vehicle data, traffic light timing and other methods [195, 46, 179, 110, 155], the vehicle's velocity along the route can be estimated as well.

To this end, employing the time-location estimation, vehicles can derive their prospective network availability and characteristics using connectivity maps [166, 154, 107, 103], covering a mapping to each location of network availability, technology and performance indicators. Connectivity map creation might be improved using advanced network monitoring mechanisms [202, 177, 178]. Since these maps cover average long-term information, it is reasonable to enhance them with short-term prediction. Proven techniques are summarized from Bui et al. [27] and include geographic and temporal pattern recognition of network load from probing data or network provider data [186], parameter gradient analysis [153] and more. The techniques consume information from various sources, and their combination may strongly improve prediction. The prediction provides a base for the algorithms investigated in this dissertation but is out of its scope because, as presented, there already exists a variety of advanced methods that probably provide sufficient quality, especially when applied in a combined manner. In Chapter 5 we even evaluate the impact of prediction errors on our algorithms and derive indicators for insufficient prediction quality.

Connectivity maps provide a good estimation for prospective transmission performance that may be enhanced with short-term predictions based on collaboratively monitored data.

2.4 CONSTRAINED OPTIMIZATION

Transmission planning approaches developed in this dissertation represent strategies, which optimize the developed transmission rating functions including constraints. Therefore, we give a brief overview about constrained optimization in this Section. Optimization techniques target to minimize a cost function or maximize a utility function, which assesses the quality of a result. Constrained Optimization Problems (CSP) additionally employ constraints that declare certain regions of the solution space as infeasible.

An exhaustive assessment of all solutions always leads to the optimal result but requires much time. Hence, methods like Branch&Cut [147] employ induction techniques to detect regions, which cannot contain results better than an already known solution. It is the art of optimization to find clever methods, which discover favorable regions fast and find their desired extrema. In our algorithms, we especially employ search-based methods. Two extreme approaches are depth-first search [204], exploring a single path and selecting in each step always the opportunity which appears to be the best, and breadth-first search [28], exploring all paths at the same time, representing an exhaustive approach. Depth-first search converges fast to a probably good solution without checking for other options. In contrast, breadth-first search explores each option, mimicking exhaustive assessment, which always leads to the best solution but requires much more time. It is the art of optimization heuristics to find a way in between, exploring the probably best paths by selecting an appropriate search order. Accordingly, optimization techniques cover always a trade-off between solution quality and execution time, i.e. whether to explore first in width or first in depth.

A popular heuristic is A* [89], which employs a quality estimation from the current explored state to the final solution. It adjusts the *search order*, deciding which path to continue exploring, taking the already explored part and the heuristic estimation for the remaining part into account. However, for many optimization problems, the quality of the solution space regions cannot be estimated properly. Hence, common approaches like tabu search, simulated annealing, genetic or evolutionary algorithms [42] use controlled randomization to explore the solution space, statistically finding regions with better extrema. A higher randomization or systematic exploration extends the search breadth and increases the probability to find a region with a better solution while a lower randomization leads to faster convergence, mimicking depth-first search in following only close-by paths with a higher probability to reach a local minimum. In particular, for constrained methods, it is beneficial to assert the feasibility and to expect the quality of alternative paths to select from. This method is called *forward-checking* and avoids the costly necessity to revert a search step in case of constraint violation, implementing an essential speed-up technique for search algorithms. Conclusively, the design of optimization approaches require, firstly, an analysis of the solution space characteristics to select the appropriate algorithm, secondly, the development of problem-specific heuristics to improve the search order and, thirdly, a trade-off decision between convergence speed and solution quality to decide for the exploration method and breadth.

3 RELATED WORK

Selecting the right network improves the perceived performance of the user, based on Quality of Service (QoS) satisfaction. In the following, we present the state-of-the-art of two aspects in this field. Firstly, we discuss the classical network selection and identify the strengths and shortcomings of existing approaches. Secondly, we present the still young research area of transmission time selection, which aims at transmitting data at a point in time when surrounding network resources are adequate. Even though both aspects have a similar goal, the work is firmly bisected. However, both try to improve the Internet connectivity of the client, exploiting data traffic distribution in one of the two dimensions: networks or time. Finally, we analyze existing mobility management approaches, which allows a client to move data connections seamlessly from one network to another. Hence, the presence of mobility management is a prerequisite for efficiently applying network selection.

3.1 NETWORK SELECTION

In many practical systems, network selection is realized using simple handover strategies [2] without a sophisticated network selection considering the user's needs. An overview of such simple applied mechanisms is presented by Park et al. [162], presenting simple mechanisms based on signal strength, distance, movement extrapolation and historical mobility patterns. However, modern network selection mechanisms go far beyond this, reflecting strategies how to satisfy the needs of the clients. The superiority of these methods is shown by Wang et al. [212]. Hence, modern network selection targets improving the access quality by distributing data traffic selectively over multiple networks, taking the current network environment and application requirements into account. We identify four important design criteria in which existing network selection approaches differ:

C1. Controller location: Network-controlled, network-assisted, and user-controlled network selection.
C2. Rating function design.
C3. Single-homed or multi-homed approaches.
C4. Information sources about the network environment.

To identify research gaps, we discuss approaches in the following, which cover the most advanced key concepts in this area of network selection research and analyze their strengths and weaknesses according to these four criteria. We illustrate the criteria describing the network selection approach landscape in Figure 1. It shows a network environment of two network operators, one providing cellular network Internet access through cells A and B, and the second providing WiFi Internet access through the hotspots C and D. Two vehicles driving along a road use their resources, from which the right vehicle is single-homed and the left vehicle is

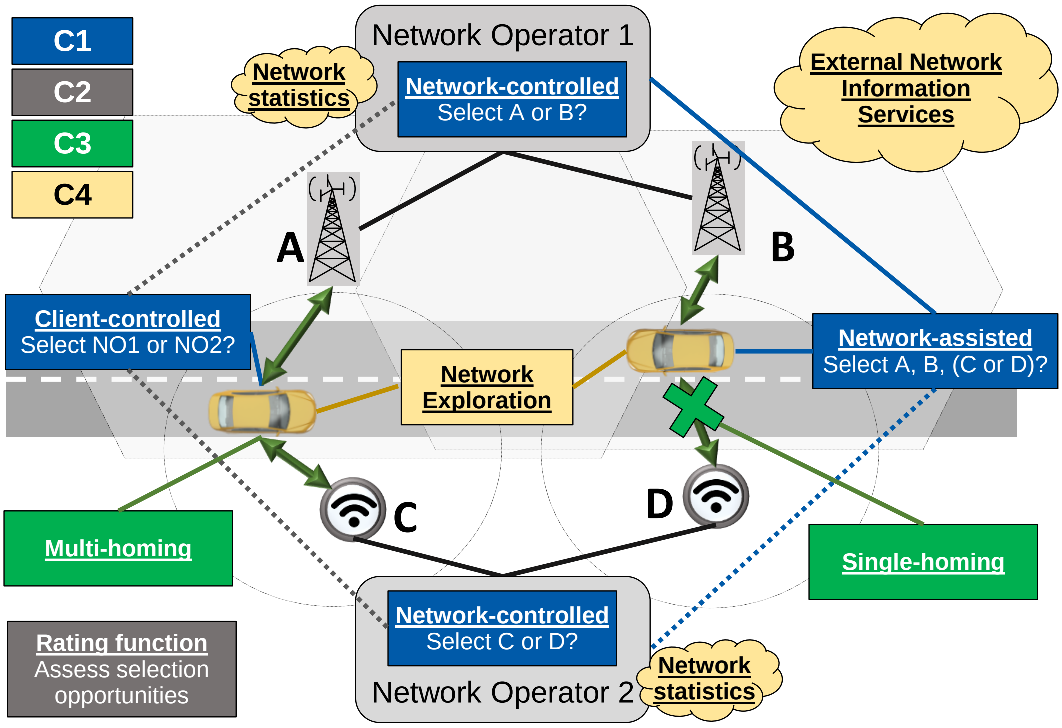


Figure 1: Network selection landscape with highlighted identified criteria. C1: Controller location (blue). C2: Rating function (gray). C3: Multi-homing (green). C4. Network information sources (yellow).

multi-homed and, able to transmit data via two networks at the same time, as identified by design criterion C3 (green). Possible controller locations according to C1 for network selection are visualized as blue boxes, located either at a vehicle or a network provider. The rating function design is not covered in the figure but mentioned as C2 (gray), whereas network information sources according to design criterion C4 are marked yellow. We define the following sections following to the first design criterion C1 and highlight the analyses on the other three criteria in the text.

3.1.1 *Network-Controlled and Network-Assisted Network Selection*

Network-controlled network selection approaches target optimization from the network operator's point of view. They neglect multi-homing and specific needs of their clients.

Network-controlled approaches, usually designed from the point of view of network-operators, see the clients as indivisible network traffic producers. Their goal is to map clients to their own access points or cells in a way, that available network resources are used most efficiently [8, 9, 21]. They focus on their own provided network resources, as exemplified in Figure 1 in the upper and lower blue box (C1), employing top-level knowledge from network statistics (C4) about connected clients to maximize resource provisioning [197, 115]. However, network-controlled approaches suffer from two significant drawbacks. Firstly, network selection is restricted to their own resources, ignoring cross-operator multi-homing (C3). Secondly, they ignore application-specific transmission requirements in their decisions, focusing solely on throughput maximization for their clients. However, as detailed in the following paragraphs, efficient network selection should consider these requirements in their rating functions (C2) to exploit the full potential of available networks. Therefore, network-controlled approaches are out of the focus of this dissertation.

Network-assisted approaches enrich the client's information base for the network selection decision with information from external databases, improving efficiency and convergence to global selection optima.

In network-assisted approaches, clients select their network based on received information about current network characteristics (C4). For Information gathering, different ways have been standardized, e.g. the Access Network Discovery and Selection Function (ANDSF) [69] from ETSI 3GPP or the IEEE Media-Independent Handover Information Service (IEEE 802.21 MIH) [85]. These standards define interfaces to request information from central and decentral sources, i.e. an external information server or the network operators, visualized as yellow clouds in Figure 1. Many approaches propose to employ these information services in theory [8, 41, 145, 49, 188], while practical approaches ignore them due to missing databases and high overhead, which is tried to be avoided.

An extreme case for using minimum information is the approach of Tian et al. [207], in which clients and networks periodically exchange a QoS satisfaction factor, which influences a probabilistic trigger of clients to re-select their network. In the case of a low QoS satisfaction of many nodes in an area, a probabilistic network re-selection is triggered for local nodes, even for satisfied ones, in order to re-distribute network resources until better overall satisfaction is reached. The selection itself stays in full control of the client, but the trigger decision is assisted from the networks and fosters across-client optimization. As exemplarily shown by Tian, network-assisted approaches can help optimizing network performance of clients by the introduction of a macroscopic view similar to that of network-controlled approaches without limiting the scope to network resources of a single provider.

An approach, in which the network operators takes an active role (C1) is presented from Khan et al. [112], integrating two relevant components: Firstly, Khan diversifies data traffic of clients by introducing traffic classes with individual QoS requirements, employing the ITU-2000 traffic classes: conversational, streaming, interactive and background, which today is the best practice in this field. To rate their transmission quality (C2), he applies a utility function, considering for QoS requirement *satisfaction* as well as monetary cost. Secondly, the authors introduce multi-homing (C3) into network selection, which allows applications with contrary QoS requirements to transmit data in parallel using different networks. Khan employs a network-assisted selection, based on reverse auctioning, in which, first, network resources are offered from network operators, providing network information according to C4, second, clients publish their transmission requirements (C2) and, third, network operators (C3) bid for being selected for their transmission. Even though the approach presents the reverse auctioning mechanism as the main contribution, we consider the developed transmission *rating* model, based on satisfaction of application QoS requirements, as well as employing multi-homing, as more important design milestones.

Rating functions targeting requirement satisfaction instead of best-effort transmission provide substantial performance benefits.

3.1.2 *Client-Controlled Network Selection*

Client-controlled network selection approaches focus on the specific data traffic of the client and available networks with their characteristics. Simple client-controlled network selection schemes assume in their rating functions (C2) that all users want to send as much homogeneous data as possible. This simplification allows investigation of strategies, in which all users can get in total the most throughput from the available networks. An example is the approach of Malanchini et al. [142], presenting a game theoretic approach, and showing that certain dynamics in distributed network selection algorithms are required to converge to a Nash Equilibrium, which equals the optimum solution. A very similar approach was presented by Zhu et al. [223], using Bayesian evolutionary games.

However, models focusing on dedicated QoS requirements of applications dominate current research activities. Even though many authors emphasize the importance of a well-defined rating metric (C2), they fall back to simple models for evaluation [55, 172, 190, 212, 217, 223]. They present the dynamics of their network selection approaches without investigating sufficiently how the discovered dynamics rely on the design of their rating model (C2). Indeed, this relation is substantial as shown by Wang et al. [214]. Most authors agree that simple linear rating models are sufficient to select the best matching networks, providing responsive decisions. Therefore, basic Multiple Attribute Decision Making (MADM) algorithms like Simple Additive Weighting (SAW), Gray Rational Analysis (GRA) and Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS), compared in [212, 39], dominate the solution landscape and show good results. They apply linear weighting and normalization to multiple attributes. GRA and TOPSIS additionally apply a direct comparison between the different results to analyze the significance of the performance surplus. Using these mathematical models, it remains a key challenge to select a proper parameter set and to design adequate utility models.

Linear decision functions are favored due to an adequate performance and a high responsiveness. A high level of detail in requirement models is important to satisfy them correctly.

One key model parameter characteristic in network selection is application QoS requirement satisfaction, linked to Quality of Experience [111, 72]. The user experiences the performance of the Internet connection only implicitly through application behavior. Whenever the applications react as expected, the characteristics of the Internet connection have been sufficient. Therefore, only adverse effects, i.e. a stalling video through application requirement violation, influence the perceived quality of the Internet connection. Hence, it is reasonable to apply a satisfactory metric. As soon as all performance requirements are satisfied, a performance surplus has no measurable benefit [160].

With good reason, many approaches use such a satisfactory metric in their rating models (C2), e.g. [161, 190, 86]. However, most also keep their models too simple to reach applicability of their rating metrics for general data traffic (C2). In contrast, they focus on a subset of data traffic, e.g. video data [129].

Considering the information source, according to C4, on which decisions are based, pure client-controlled network selection algorithms have to explore networks themselves, illustrated by the yellow box in the center of Figure 1 and have to cope with limited information. A simple method from Senouci et al. [190] introduces belief functions, which reduce the weights of different criteria according to the information quality in their MADM-based approach. Hence, networks for which a more accurate characteristic prediction exists are preferred over others. Although this leads to better predictable perceived characteristics, the method underestimates the potential from using unknown networks. An approach with the same goal is given in by Taleb et al. [203], who additionally propose to access the network information service ANDSF, as explained above.

Network exploration enriches the information base with current information, improving decision quality.

A method to encounter the issue of incomplete information (C4) about networks is presented by Wu et al. [217], employing online learning algorithms based on a multi-armed bandit model. Selecting an arm corresponds to selecting an available network. After selecting a network, the algorithm uses it for at least some test samples, probing its quality, updating its reward functions for this network and, again probabilistically selecting a network based on the expected reward. This algorithm can adapt well to changing environments and changing data traffic. However, each environment change leads to an exploration phase of a few hundred milliseconds duration in which handover processes impair the transmission quality. Du et al. [56] present a similar approach using Q-learning, which seems to converge slightly faster.

3.1.3 *Conclusions*

Network-controlled network selection strategies optimize the performance of managed networks efficiently and are common practice. However, they exclude network resources from other operators or non-managed networks, e.g. private WiFi access points, from network selection. Hence, we focus on user-controlled strategies. Moreover, we do not consider the two as competing approaches, but rather as complementary ones, with client-controlled strategies selecting between optimized operator-networks.

Distributing data traffic over those managed networks improves the perceived transmission quality substantially. We learned that the proper design of the *rating function*, according to C2, is essential for network selection. Authors that compare different selection strategies agree that increasing the level of detail for application requirements in employed rating models leads to substantial benefits [142, 217, 214]. Especially a network selection, which relies on *application QoS requirement satisfaction*, leads to advantageous decisions [16, 111].

Dealing with *incomplete information* about available networks, addressing C4, is addressed in two ways: Firstly, exploring the network performance and, secondly, accessing remote information, both improving the local information base. While remote information can significantly decrease the need for inefficient exploration, it might still fail in the presence of non-average data traffic conditions. Thus, exploration is required to react to sudden changes in the network performance. A combination of both is useful. Accordingly, *adaptive approaches* seem to be most promising, which use external information bases and, in extension, adapt the selection based on actual environmental conditions.

Moreover, only a few approaches integrate *multi-homing* in their strategies, addressing C3. Enabling the access to multiple networks in parallel significantly boosts the performance in application QoS requirement satisfaction, as shown in

[112]. Hence, contrary requirements of different applications can be satisfied at the same time using two or more networks in parallel.

Conclusively, related works identify critical design space parameters for network selection strategies but barely focus on the underlying rating functions. For client-controlled approaches, we consider satisfactory rating with a detailed application-specific model as mandatory, while the presence of multi-homing provides enhanced selection opportunities. A similar combination was only considered in a strongly network-assisted approach of Khan et al. [112]. In the next section, we discuss an additional selection component, which the presented approaches do not examine at all: transmission time selection.

3.2 TRANSMISSION TIME SELECTION

For moving clients, the perceived network environment changes rapidly with their location. Especially small and medium range networks get into and out of reach frequently, while perceived characteristics of wide area networks change as well.

Transmission time selection seeks for good opportunities to transmit delay-tolerant data, e.g. when WiFi is available.

According to the idea of network selection, application dependent choice of networks is essential. Hence, for each delay-tolerant data transmission, a selection of the transmission time is important as well, as discussed in the current section. We identify two important criteria in which time selection schemes differ:

C1. Considered planning time horizon.
C2. Rating function design.

In the following, we discuss selected approaches that illustrate the current state-of-the-art in time selection. Furthermore, we discuss their strengths and weaknesses regarding the criteria above.

3.2.1 *Relevant Approaches*

Bui et al. [26] improve video transmission through time selection using the mobile network only. Hence, they present a video transmission framework, predicting the future mobile network access of a client. Based on this prediction, they estimate the point in time when to fill up buffers to optimize transmission cost. However, they assume the cost to be anti-proportional to the current mobile network capacity. Hence, clients should preferably transmit data when the network capacity is high. They split the forecast model into a short term and a medium/long term phase. For short term forecasting, they apply simple auto-regressive filters and select the parameters according to the user mobility. In contrast, the medium/long term forecasting models take adjacent cells and movement into account. Hence, they combine information from network exploration and external data sources to estimate the network performance. As discussed in the previous section, we consider this as a beneficial approach. Based on the network estimation, they propose an algorithm that fills the video buffer when the network capacity is expected to be highest. They state that large buffers help to make this method efficient because they enable higher delay-tolerance of data transmission. Furthermore, they reach a 30% cost gain over an algorithm that simply fills up the buffer whenever possible. Even though Bui's assumption that cost is lower when network capacities are higher are hard to defend, the approach makes promising points. To clarify the benefits, we analyze their approach using the findings of Lu et al. [137]. They propose an algorithm similar to that of Bui et al., sending data when network capacity is high, aiming for decent video delivery. They extend the analysis by a multi-user scenario. Focusing on the interplay of multiple users that compete for bandwidth, they show that the presented strategy increases overall user satisfaction. Lu proves that the approach statistically reduces critical load peaks, especially when there are few network resources. This avoidance of critical peaks increases the probability that application QoS requirements are satisfied at any time.

Transmission in bursts using high-throughput networks or WiFi networks reduces transmission time to save energy and releases resources during phases with low network performance for delay-sensitive transmissions.

Indeed, the method leads to short high throughput bursts in high capacity networks. This transmission pattern has been identified as the most energy preserving in the analysis of Lee et al. [132]. It allows the client to put its interfaces to sleep in transmission pauses and, hence, to save energy. In contrast to Bui, he considers using also WiFi networks for data transmission. In fact, he considers a scenario of traveling in the subway of Seoul, where mobile networks are available. At each station, there is additional free WiFi available. This scenario does not need a sophisticated prediction strategy because its time schedule is deterministic. Using optimal offloading policies with large buffers, Lee reaches up to 63% cost reduction through WiFi offloading and conserves up to 42% of energy for communication. This performance gain shows that with an accurate knowledge about future network characteristics, delay-tolerant data transmission can be efficiently moved to better suited time and networks.

Moreover, the idea to use WiFi to complement the Internet access is discussed in the following approaches. The framework Wiffler from Balasubramanian et al. [15] follows this goal to offload delay-tolerant data. The study unveils two interesting insights: Firstly, WiFi complements 3G availability, highlighting the importance to use both in a smart way. Secondly, for the vehicle use-case, about half of the data traffic can be offloaded to WiFi using a maximum delay-tolerance of one minute. A detailed analysis of the impact of delay-tolerance on the offloading ratio and the energy saving is presented in [133] by Lee et al. They conclude that 64-87% of data can be sent via WiFi when offloading with a delay of up to 6 hours. They expect an energy saving for transmission by 50-75%, confirmed by other analysis. Energy saving in delayed offloading was also considered Ra et al. [168] including the factor of network data rates in deadline calculation.

These approaches confirm that transmission characteristics can be improved using time selection. Furthermore, the benefit increases with the magnitude of delay for offloading. Using larger buffers in video delivery means that data can be prefetched for a longer duration. In fact, many approaches treat the tradeoff between monetary cost and time, as presented in the following. Waiting longer for data increases the potential for delayed offloading. While bufferable video data is delay-tolerant, this does not hold for much other data traffic. Waiting for transmissions to complete, hence, is uncritical for, e.g., bufferable streams and background data. In contrast, the user is impatient for other data transmissions, e.g. interactive data requests or conversational traffic. The framework AMUSE [99] from Im et al. exploits this potential, characterizing data traffic and asking the user for feedback. The approach plans transmissions for a complete day, adapting the plan gradually. They stress the finding that efficiency of these approaches rises with higher delay-tolerance and more delay-tolerant data. Thus, their approach focuses on identification of the magnitude of delay-tolerance of each data transmission. Therefore, the authors do a survey and derive parameters about the willingness-to-wait for different application data. Consecutively, they derive a model for the user's willingness-to-wait of each application, incorporating on data amount and preference. These model terms are used in a linear MADM rating function, as explained in the previous chapter. Applying their approach, they reach a cost reduction of up to 36%. The limiting factor willingness-to-wait is also identified by other researchers. To increase the offloading potential, Cheng et al. [40] propose to offer incentives for data offloading. Hence, they want to convince the users to wait longer for data transfers, especially during network capacity shortages. They summarize different strategies, ranging from extending high-speed data contingents in mobile networks to monetary cash backs [224].

Time selection approaches focus on delay-tolerant data only, ignoring mixed data traffic. Network selection is reduced to WiFi-preferred and satisfying transmission deadlines.

The presented approaches show that a significant amount of data can be delayed for later transmission via WiFi. They follow, like many other researchers, one assumption: WiFi is always the best. However, this is not the case. Especially in congested network, the service quality might decrease significantly and not allow applications to transmit data to their satisfaction. Cheung et al. [41] present an approach, claiming to be the first ones incorporating QoS requirements of applications into the delay decision. However, they limit their QoS rating function to one single parameter: transmission deadlines. Remembering the key attributes of a network selection algorithm, as stated in the previous section, Cheung cannot hold their promise. Nevertheless, we see Cheung's claims as a first step in the right direction. It considers that WiFi is not always the best option even when available. A similar conclusion from analyzing deadlines is drawn by Mehmeti et al. [144, 145]. However, they also limit the network selection decision to holding a deadline.

3.2.2 *Conclusions*

There is a research gap for a combined strategy treating delay-tolerant as well as delay-sensitive transmissions, exploiting the benefits of network selection and transmission time selection at once.

The presented approaches clarify the potential of selecting the time for data transmission. Even when only a single network is available, a good temporal transmission pattern can improve the performance, minimizing waiting times and saving energy. Most approaches conclude that the benefits of temporal offloading rise with the amount of data to be offloaded and the limit of the potential offloading delay. However, the full potential of time selection evolves when considering additional networks and client mobility. The continuous location change of the moving client comes along with a rapid change of the perceived network environment. State-of-the-art delayed offloading approaches do not exploit the full opportunities. They aim to offload as much data as possible to WiFi, disregarding application QoS requirements. Hence, they design unbalanced rating functions, which reflect indistinct goals yielding to simple WiFi-preferred strategies.

To the best of our knowledge, nobody did focus on creating rating models or transmission strategies, which completely integrate the advantages of both, transmission time selection and network selection, in order to exploit the full potential for data transmissions in heterogeneous wireless networks. We target this research gap in Chapter 4.

3.3 MOBILITY MANAGEMENT

To realize smart data flow distribution over networks and time, a network protocol is required that is able to shape data traffic and route it flexibly. These challenges are treated in the research domain of mobility management. To highlight the differences in state-of-the-art mobility management approaches, we do not focus on dedicated mechanisms but stress particular design decisions in them. Most approaches follow similar patterns and peek out only in a dedicated design decision. In the following, we describe the concept of mobility management highlight outstanding design decisions of selected protocols. Thus, we present the highlights and shortcomings of more than 15 years active and still ongoing mobility management protocol research and condense them into an easy to grasp discussion.

3.3.1 *Problem Statement and General Concept*

Mobility management pursues continuing network connections, even tough the point of presence of the client changes, e.g. when it connects to another network. This is called a handover. The essential problem in handover processes is the change of the client's IP address in the network layer. Transport layer protocols like TCP and UDP rely on the IP address, storing them locally as a reference while the connection is alive. In fact, they use the pair of the source and destination IP addresses to identify the connection. What happens when an IP address changes? We assume the client to change its IP address as a result of a handover. All network packets are sent now via the new point of presence and therefore carry a new IP address. The packets with a new IP address arrive at a server, which tries to identify to which connection the packets belong. Therefore, it compares the source and destination IP address of the packet to those of the active connections in its local memory. However, it is not able to find the new pair, since the source IP address has changed. Accordingly, it discards the network packet. Consecutively, the connection runs into a timeout and breaks. To continue the transmission, the client has to re-initiate the connection.

IP addresses are employed for routing and connection identification, forbidding to change the routes of running connections. The general solution is a separation by introducing a new persistent identifier.

This problem originates from the *double role of IP addresses*. IP addresses serve, firstly, as a so-called *locator*. They identify the location, i.e. the point of presence, of the client and are used for packet routing. Secondly, at the same time, the IP address serves as an *identifier*. It is used to identify the client, i.e. its active transport layer connections.

The general solution for this problem is to introduce a new identifier for active connections, in order to separate the roles and let the IP addresses serve as locators only. The mobility management protocol is then in charge to provide a dynamic binding between locator and identifier, as illustrated in Figure 2. In the following, we analyze how mobility management achieves this and how certain design decisions affect their non-functional properties. The following state-of-the-art analysis is partially based on our publications *A Concept for Vehicle Internet Connectivity*

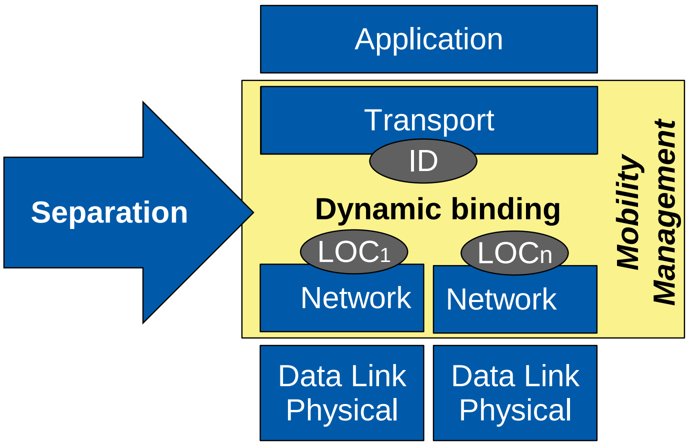


Figure 2: Mobility management general concept

for Non-Safety Applications [184] and *MoVeNet: Mobility Management for Connected Vehicles* [182] but covers extensions and modifications.

3.3.2 *Direct and Proxy-Based Approaches*

An essential difference of mobility management protocols is whether they use proxies or not. Hence, we explain different proxy schemes and their advantages and disadvantages in the following. Proxy schemes use one or more intermediate nodes for packet forwarding, while control packets are only sent to the proxy. Using proxies introduces protocol transparency, round trip times and handover latency. In contrast, direct handover schemes set up a straight communication between a mobile node and its communication partner. Data as well as control messages are exchanged only between these two entities. In the following, we discuss the three options for direct, single-proxy and multiple-proxy mobility management schemes. We illustrate the three schemes in Figure 3.

3.3.2.1 *Direct Mobility Management*

Direct mobility management suffers from repeated initialization overhead and incompatibility to conventional network stacks.

In direct mobility management schemes, client and server exchange data and control information using the direct route. Thus, both entities have to implement the mobility management protocol in order to use the mobility features. Accordingly, the feature is restricted to those servers in the Internet, which implement the modified network stack. The protocol is not transparent. It cannot be used from nodes using a conventional Internet communication stack. After the connection setup, the protocol context must be established to start protocol operation. This often requires several round trip times and forbids using the protocol right after connection start. Accordingly, using direct mobility management protocols is only beneficial for heavy tailed data connections that persist for a longer duration [156]. Furthermore, each control signaling requires at least one round trip time for request and acknowledgment. In addition, setting up the protocol context for each individual connection creates a certain overhead. However, the big advantage of the direct protocol scheme is that it does not rely on additional entities. Therefore,

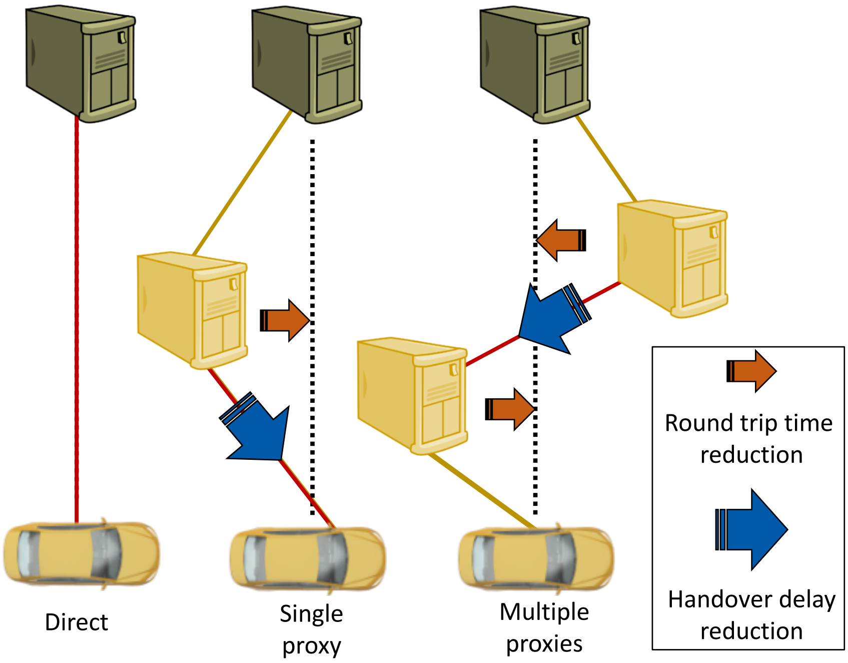


Figure 3: Direct mobility management (left) versus proxy-based mobility management (center, right)

they are simple to introduce for dedicated clients and servers even though they lack compatibility to legacy servers in the Internet. Relevant protocols of direct mobility management are Multipath TCP [73], Host Identity Protocol (HIP) [151], Shim6 [159, 170, 169] and Multipath SCTP [191].

3.3.2.2 *Proxy-based Mobility Management*

In contrast to direct mobility management schemes, proxy-based mobility management introduces one or more intermediate nodes into the communication path, resulting in two essential advantages. Firstly, a proxy can provide protocol transparency. The proxy hides the mobility management protocol operation and converts the data flow into a conventional connection towards legacy nodes. Hence, the mobile node can communicate with all servers in the Internet. This design is prevalent in Mobile IPv6 [164]. However, the idea to reach protocol transparency through a proxy has also been applied to Multipath TCP derivatives [158, 171].

A proxy introduces protocol transparency for communication partners. Thus, it ensures compatibility with the existing Internet architecture.

Due to adding a second proxy, the mobility management protocol can even be hidden from both nodes, the client and the server. In this case, the intermediate system manages client mobility completely. Protocols following this concept are Proxy Mobile IPv6 [83], the Locator-Identifier Separation Protocol (LISP) [71] or Mobile IPv6 Network Mobility (NEMO) [48]. The first and second select the access points as a proxy, identify the client within a managed network and update its association at the proxy. Thus, the client is not involved in the handover process. Therefore, the handover happens only between the access point and the mobility managing proxy. Since they are interconnected with a backbone cable network, handover latency is very low [129]. In contrast, the last approach is meant to manage a whole subnetwork at once. For example, clients in a train connect to a NEMO router. The router acts as a Mobile IPv6 capable client managing the mobility and provides persistent IP addresses to customers [38]. Accordingly, all three approaches significantly reduce signaling overhead via the wireless link. Combining the approaches, using Network Mobility for Proxy Mobile IPv6 schemes reduces overhead further [100].

The second advantage of proxy schemes is a reduced handover delay. A handover determines the switch of a client from one access point or cell to another. A small handover delay enables fast reaction on sudden changes. During a handover process, the client can usually not send or receive data. Hence, a short handover delay is beneficial. In mobility management protocols, the handover delay is determined from the latency between the triggering and the executing node. To reduce the handover delay in proxy schemes, the proxy should be as close as possible to the triggering node. In Figure 3, we marked those lines in red and attached a big blue arrow to signalize how proxies should move towards the triggering node to reduce handover delays. An approach, which exploits the closeness of proxies for handover latency reduction, is Hierarchical Mobile IPv6 [196]. It introduces a second level of proxies close to the mobile node in order to reduce the distance and, hence, the delay between them. As soon as the mobile node has traveled a longer distance, the main proxy switches to a closer second level proxy to keep the optimized distance small [194]. However, the performance of this approach is limited by the dominant delay over the wireless link. The above-mentioned approach Proxy Mobile IPv6 takes this delay out of the signaling loop and, thus, reaches significantly lower handover delays. However, handovers are restricted to the local domain of the network operator [130].

The proxy location is crucial for handover latency and packet round trip times. Proxies should be close to the client and close to the optimal route.

Proxy servers create certain detours that increase the round trip times for data packets. To reduce round trip times, proxies should be close to the ideal path to the communication partner. We visualize this in Figure 3 using orange arrows, indicating that proxies should in best case be located on the direct route between the mobile node and its communication partner without introducing routing detours. Direct handover schemes do not share these problems. Inspired by this fact, there exist route optimization mechanisms, which omit the proxy use [11, 43, 91, 14]. However, this concept violates the transparency aspect and reduces proxy-schemes to direct mobility management schemes. Therefore, they have been replaced by the novel concept Distributed Mobility Management (DMM), as detailed in the following.

3.3.2.3 *Distributed Mobility Management*

An upcoming trend for round trip time reduction arises from an ambiguity when multi-homing is active, i.e. the client uses multiple network interfaces in parallel. We visualize this ambiguity in Figure 4. The left side shows the single-homed case of a client, served from a single network operator. The client communicates with multiple servers. To reduce the packet detours for all servers in the same way and additionally decrease handover latency, the proxy should be as close to the client as possible. Accordingly, Mobile IPv6 proposes to use the first access router as a proxy [164]. However, when using the networks of multiple operators at the same

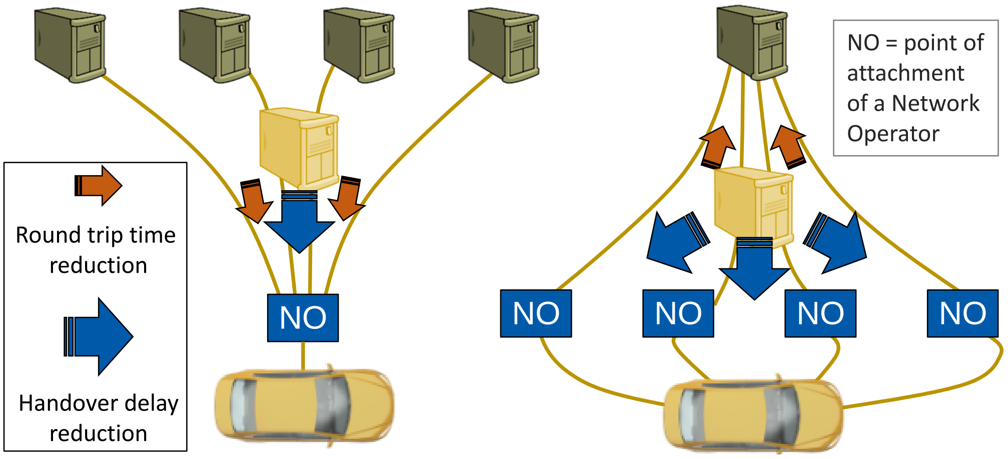

Round trip time reduction
Handover delay reduction
NO
NO
NO
NO
NO
NO = point of attachment of a Network Operator

Figure 4: Ideal proxy location for single network operator Internet access (left) and for using multiple network operators (right)

time, the picture changes as illustrated in Figure 4 right. For a single connection, the proxy should be closer to the communication partner to reduce packet detours, i.e. additional hops along the routes. Furthermore, the proxy should be close to the used network operators' points of presence to the Internet to reduce handover latency. Accordingly, the proxy location for each connection should be close to the points of presence of the most used network operators and also close to the optimal route to the communication partner.

Distributed Mobility Management is an answer to the proxy location problem. It uses multiple proxies, which can adopt close to optimal proxy positions for individual data flows.

To solve this ambiguity, the concept of Distributed Mobility Management was introduced [226, 76, 150]. Instead of using a single proxy for all connections, each connection may use a different proxy. The proxy selection can be made based on the communication partner's location and the expected used network operators in order to reduce round trip times. Moreover, there exist two different management methods of this distributed proxy architecture: Fully and partially distributed management [78, 130]. In fully distributed mobility management, proxies have to organize themselves and can share information with each other [63]. Each proxy is able to provide the mobility management service alone. In comparison, partially distributed approaches have an additional central instance, inspired from a software-defined network controller, which acts as end-point for signaling traffic and can organize the interplay of proxies [77].

3.3.2.4 *Layer of Mobility and Handover Granularity*

Mobility management protocols can be implemented on different network layers [57], c.f. Figure 2. As marked in Figure 2 in the yellow box, mobility management is usually managed within the network layer or the transport layer.

Application Layer: Many applications implement mechanisms to continue their operation after network failures, e.g. Skype, including a switch of the network access. However, these mechanisms require running into a timeout before they are able to recognize the change. Hence, this kind of mobility management is not seamless. In addition, each application has to implement and run the mechanisms on its own which creates redundancy. Therefore, we do not consider these approaches further.

Transport Layer: The transport layer covers flow control and is, therefore, well suited for mobility management. Flow control enables shaping of the traffic during and after the handover. Moreover, all relevant information is available in this layer, covering IP addresses and ports. This enables routing of individual data flows via different networks and also flow forking, which distributes packets of a single data flow over multiple interfaces. Examples are Multipath TCP [73] and mSCTP [117, 14]. However, the disadvantage of these approaches is that they are restricted to the one protocol. Managing not all data traffic introduces a load balancing uncertainty already for the client side. This uncertainty might not be relevant for handover itself only, but it conflicts with the goal of this thesis: smart data flow distribution over networks and time.

Network Layer: Network Layer implements the Internet Protocol (IP) and forms the so-called waist of the Internet. It covers all data traffic, reflecting a significant advantage for mobility management. However, it does not allow identification of individual data flows. Accordingly, Mobile IPv6 approaches do not consider routing of individual data flows via multiple different networks. However, this has been recognized as an important feature. Extensions have been developed in [209], extracting information from transport layer for flow identification.

Routing-based: Another way to control packet routing, which gains importance, is Software Defined Networks (SDN). A controller optimizes forwarding rules for a network of 'dumb' routers in order to optimize transmission characteristics and balance loads over multiple data routes. SDN can be applied to realize mobility management [113, 84, 76, 79]. However, SDN suits mainly for optimization within a controlled network, as operators may not allow external control of their hardware. Hence, it matches to network-controlled mobility management schemes. SDN-based mobility management can reach high performance in a completely controlled subnetwork, even without modifying the underlying architectures. This qualifies the solution for intra-operator use.

3.3.3 *Conclusions*

Recent discussions within the area of 5G design brought a new focus in mobility management towards heterogeneous access networks, motivating for low-latency connections and mechanisms. These discussions resulted in new Distributed Mobility Management (DMM) concepts, further enhanced by the control plane separation concept from Software Defined Networks. Current research in DMM focuses completely on network operator-controlled concepts, which permit lowest handover latencies and lowest overhead to the mobile node. However, these approaches ignore coupling of access networks beyond the managed domain, e.g. complementary cross-operator use of multiple cellular access networks and WiFis. A different research area for mobility management developed with the rise of Multipath TCP (MPTCP). It focuses completely on enabling this complementary network use but ignores the results from earlier mobility management research, neither reducing handover latency or management overhead nor providing compatibility to all IP traffic or legacy backend nodes.

Existing approaches suffer from individual shortcomings. There is a research gap for an approach satisfying the requirements of the connected vehicle scenario.

The automotive scenario reflects a clear demand for a concept, providing the advantages of both approaches, firstly, enabling comprehensive mobility management in multi-operator heterogeneous networks and, secondly, providing optimized connection and handover characteristics as well as applicability to all data traffic, even to legacy nodes in the Internet. To fill this gap, we propose our client-controlled protocol Mobility Management for Vehicular Networking (MoVeNet) in Chapter 6, inspired from Distributed Mobility Management but independent from operator entities of access networks, which integrates the demanded advantages. However, we do not see it as a replacement of network-operator-controlled concepts but as a complementary enhancement mechanism, providing mobility management on a higher scale covering all IP access networks. Thus, operators should optimize characteristics within their managed network as proposed from existing DMM methods, while, on top of this, MoVeNet glues managed as well as non-managed IP access networks together, providing a comprehensive and flexible Internet access for mobile nodes.

4 STRATEGIC DATA TRANSMISSION

Data transmission for mobile nodes significantly benefits both from reasonable wireless network selection and from selecting a proper transmission time, as shown in the related work analysis in Chapter 3. However, their combination has not been sufficiently considered. In particular for scenarios with high mobility, as the connected vehicle scenario focused in this dissertation, a combination of both seems promising, defining the objective of this chapter.

Existing approaches for network selection usually restrict their scope to the present-available networks and present-available data flows, i.e. those that are available at the current moment. Clients profit from distributing the data traffic reasonably over the available networks. This leads to a network selection in which for each application the requirements are tried to get satisfied in the best possible way. However, the restriction of these approaches to *present-available* networks is an arbitrary simplification. In contrast, scientific work on delayed offloading demonstrates the benefits of selecting the proper transmission time slot for delay-tolerant application data. However, these approaches restrict their network selection to simplistic models, as WiFi preferred, and delay the data transmission, if possible, to match the temporal availability of WiFi networks. Hence, both research directions demonstrate an essential optimization potential for distributing the data traffic of a mobile client over one dimension: networks or time. Nevertheless, to the best of our knowledge, nobody combines both in a proper fashion. A detailed analysis of the related approaches is given in the Sections 3.1 and 3.2.

The goal of this chapter is to design a joint time-network selection and to analyze its effects on the *perceived* transmission performance, i.e. focusing on QoS violations that lead to effects which are visible to the user, like a stalling streamed video or long delays during a conversation. Therefore, we target the integration of network selection into transmission time selection. To combine the two concepts we, firstly, define a combined rating model, presented in Section 4.1. It generalizes specific rating model components from network selection, covering classical QoS parameters, and model components from time selection, like deadlines, and extends them with a novel transmission requirement model that allows to treat all data in a unified way. The transmission rating model represents our first main contribution. Secondly, in Section 4.2, we develop a combined time-network selection approach, which we call Joint Transmission Planning (JTP), defining our second main contribution as a novel approach for explicit network selection integrated into transmission time selection. It employs heuristics for flow prioritization and network selection derived from the presented transmission rating model. For evaluation, we compare JTP to three predominant state-of-the-art transmission strategies, employing the same heuristics for a fair comparison of the strategies. We show in Section 4.3 that JTP delivers a robustly high performance of 87-91% of the scenario optimization potential, significantly outperforming compared approaches by 7-25%.

This chapter is based on our publication *Impact of Time in Network Selection* [183] and extends the therein presented work with model improvements and advanced analyses. Note that the employed time selection relies on a prediction of prospectively available resources. We exclude the effects of prediction errors in this chapter, assuming complete information about future-available networks, their characteristics and the application data to be transmitted over time. The impact of prediction errors on transmission planning and ways to handle them are addressed in Chapter 5.

4.1 TRANSMISSION RATING MODEL

The perceived transmission quality targets transmission-specific components of QoE, ignoring system and context components.

Our Transmission Rating Model presented in this chapter assesses the *perceived transmission quality*, which is a subset of the often employed Quality of Experience (QoE). According to Reiter et al. [173], QoE includes influence factors related to the system, the user and the context. It measures the delight or annoyance of using an application or service using a subjective mean opinion score. To define the perceived transmission quality, we limit these influence factors to that related to Quality of Service (QoS) metrics and monetary cost, as these are the parameters that can be addressed from the transmission only. Accordingly, our rating model provides a quantitative measure of application QoS satisfaction, balanced it with monetary cost, reflecting conflicting satisfaction indicators of users. To this end, we integrate the models from time selection into network selection into our approach. An overview about all model parameters and variables is shown in Table 2 on page 41. To show the structure of the model, we firstly present the model components and explain the most important components and weights. Secondly, we detail their interactions and constraints in a formal model definition. Before we explain the model components in the following, we first present two granularity abstractions as prerequisites.

The model abstracts over time using time slots and data amount using data tokens. Both can vary in their granularity. If the duration of time slots is long, a planning time horizon is covered with fewer time slots. This corresponds to a low granularity in time. It reduces the computational effort for planning and constitutes a trade-off between accuracy and execution time, which we study in detail in the evaluation. We divide the time τ in time slots t of equal duration $\Delta\tau$. Hence, time slots are defined by Equation 4.1.

$$t = \frac{\tau}{\Delta\tau} \tag{4.1}$$

Time and data amount are defined abstractly as time slots and tokens of parametrized size to allow variation of planning granularity.

Equivalently, the data amount can also be modeled with different granularity. Thus, we define the data amount d_f of a data flow f as a number of tokens to be planned $\widehat{p}_f$. Furthermore, the data rate $d_{t,n}$ of a network n in time slot t is defined as a bucket that can hold a number of tokens $B_{t,n}$. The two data metrics are scaled using the data token size d and are calculated using the Equations 4.2 and 4.3.

$$\widehat{p}_f = \frac{d_f}{d} \tag{4.2}$$

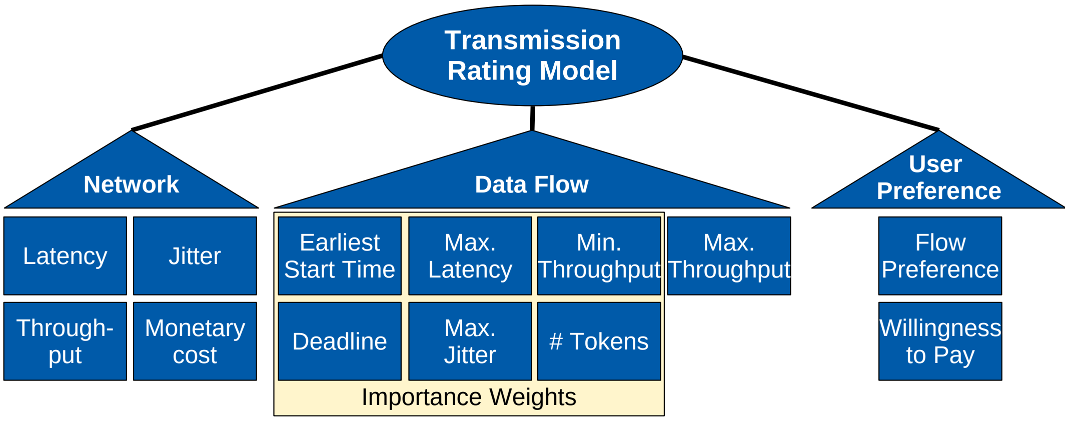


Figure 5: Transmission Rating Model Components

$$B_{t,n} = \frac{d_{t,n}}{d} \tag{4.3}$$

4.1.1 *Model Components*

The rating model consists of three basic components. Firstly, the network model, secondly, the data flow model and, thirdly, the user preference model, as visualized in Figure 5. Through the interaction of the components, the model rates a transmission plan p, which defines how many data tokens $p_{f,t,n}$ of a flow f are allocated to a network n in time slot t. We present the model components briefly in the following.

4.1.1.1 *Network Model*

We model networks with their availability over time and their transmission characteristics. Our model covers throughput, latency, jitter and monetary cost for transmission. Latency L_n, jitter J_n and monetary cost c_n^{mon} are fixed parameters for a network n over the treated time span. This network parameter set complies to state-of-the-art network selection models [208] even though many new approaches focus on throughput only, restricting their analyses to data traffic for which the throughput impact dominates. To model throughput, we reduce all states that influence network characteristics like location, network load or environmental effects to the time dimension. Hence, for each point in time, we use the estimated throughput of the client at its corresponding location and the network state. To provide that estimated throughput, the model requires a network prediction and movement prediction, as presented in Section 2.3, from which the characteristics for each time slot can be derived. As a result, we assign an estimated throughput value for each network n to each time slot t. It is defined as network capacity per time slot t in buckets with individual size $B_{t,n}$. This model goes beyond network selection approaches because it considers the prospective throughput over time, combining it with the models from time selection approaches [132, 203].

Network throughput is modeled as a number of transmittable data tokens in each time slot.

4.1.1.2 *Data Flow Model*

The data flow model defines application QoS requirements to the data transmission. A requirement of property x is defined by $\hat{x}$. The first requirement is about the amount of data of a flow f to be transmitted, defined through a number of tokens $\hat{p}_f$. Each token represents a fixed amount of data d, as defined in Equation 4.2. In addition, we let non-allocated tokens $u_f(p^*)$ contribute to the cost function. We use p^* as an alias for p indicating that in the employed function, the absence of a token allocation creates an attracting force. Further requirements of the data flow are minimum throughput, latency $\hat{L}_f$, jitter $\hat{J}_f$, an earliest start time $\hat{t}_f^{st}$ and a deadline $\hat{t}_f^{dl}$. These requirements are defined for each data flow and are attributed with individual importance weights for each property x using ω_f^x. The presence of flow-specific importance weights is signalized with the yellow box in the center of Figure 5. Data flow priorities are inherent to the specified requirements and their importance. Accordingly, a data flow with many important requirements is considered as more important than a data flow without special requirements to the transmission.

4.1.1.3 *User Preference Model*

If the inherent data flow priorities, as defined in the preceding paragraph, do not comply with the user's expectations, the user preference model may be employed for personalized balancing of flows with the weight ω_f^{user}. Furthermore, the user preference model provides a trade-off between the contradicting objectives of monetary cost and desired performance using the global willingness-to-pay parameter, i.e. the monetary cost weight ω^{mon}, which the user can decrease to improve the application QoS satisfaction at an expense of allowing a higher monetary cost for transmissions.

4.1.2 *Formal Transmission Rating Model Definition*

The model defines constraints and a rating function for a quality assessment of a transmission plan according to application QoS satisfaction and monetary cost. It forms the basis for integration of network selection into transmission time selection models. In the following, the model is presented along with the corresponding equations.

The rating model defines a linear weighted multi-objective cost function that balances application QoS violation, monetary cost and handover overhead.

4.1.2.1 *Objective*

The objective of strategic data transmission is to find a transmission plan p that satisfies the defined constraints and minimizes the rating cost function $c(p)$, as specified in Equation 4.4. The cost function consists of the application QoS violation cost $c^{flow}(p)$, the monetary cost $c^{mon}(p)$ and an additional flow migration cost component $c^{mig}(p)$ that punishes switches between networks to avoid ping-pong effects. Note that this represents a multi-objective optimization with fixed objective weighting. During the evaluation, we vary the monetary cost weight to show the effects of different objective priorities.

Minimize $c(p)$ subject to the constraints C1-C3 in Equations 4.10, 4.11 and 4.12.

$$c(p) = c^{flow}(p) + c^{mon}(p) + c^{mig}(p) \tag{4.4}$$

4.1.2.2 *Model of Forces*

The different cost function components have to be subsumed in a function. To describe their interaction, we employ a model inspired by acting forces, providing two beneficial properties. Firstly, existing forces lead to a certain tension in a system while, secondly, contradicting forces eliminate each other and the resulting force reflects an imbalance of the tension. In the transmission rating model, each cost function component, e.g. violations of application QoS requirements and monetary cost, may activate a force. The presence of forces creates a certain tension in the system, reflecting the quality of the transmission. Hence, the absence of tension, when there are no forces, complies to a high transmission quality, e.g. a transmission with insignificant application QoS violations and low monetary cost. Furthermore, imbalanced opposing forces indicate room for improvement, i.e. that a different trade-off of conflicting goals could reduce the overall tension in the model and, thus, increase the quality of the plan. For example, a minimum throughput requirement violation might be compensated by employing a maybe more expensive network with a higher data rate, if available.

In our rating model, requirement violations and other influence factors represent attracting and repelling forces that in total create a tension and an imbalance, reflecting the transmission quality.

We bisect the transmission rating model's cost function into two mutually exclusive activated components, which follow the model of forces, representing attracting and repelling forces. Accordingly, the cost function can also be written in terms of forces as presented in Equation 4.5.

$$c(p) = c^{attr}(p) + c^{rep}(p) \tag{4.5}$$

The attracting forces $c^{attr}(p^*)$ arise from non-allocation of data tokens in a plan p^* and pull data tokens towards allocation to networks in general. Their magnitude depends on flow-specific requirements, creating a general priority for flows, as detailed in the following sections. In contrast, the repelling forces $c^{rep}(p)$ arise from, e.g., QoS requirement violations and can only be active for allocated tokens in a transmission plan p. Employing the repelling forces, tokens of individual data flows push themselves away from non-matching networks and time slots. Due to cost function minimization, the effects of the forces distribute data tokens to those networks and time slots, which match best for transmission, balancing the attracting and repelling forces and reducing the overall tension. We illustrate the impact of forces in Figure 6. Their characteristics are detailed in the following sections along with the mathematical modeling of each cost function component. To give an overview, we introduce the repelling forces, followed by the attracting forces.

The monetary cost $c^{mon}(p)$ emerges from data transmission over a network and is a repelling force that pushes data tokens away from expensive networks. Note that it is time invariant over the planning horizon and, therefore, represented as a horizontal bar in Figure 6. Furthermore, flow migration cost $c^{mig}(p)$ punishes handovers between networks because they create signaling overhead and disturb the flow's transmission continuity. Hence, it is a repelling force, pushing tokens away from all networks except the currently used one. To explain the forces from the data flow model, we detail the application QoS requirement violation component $c^{flow}(p)$, as shown in Equation 4.6. It consists of a linear combination of six components, weighted for each flow with the user's preference for this data flow ω_f^{user}. Its first two components are deadline violation cost $c_f^{dl}(p)$ and start time violation cost $c_f^{st}(p)$. They contribute quadratically to the strength of violation, i.e. the time offset, as shown in Figure 6. They punish token assignment apart from the temporal desired scope. Latency violation cost c_f^L and jitter violation cost $c_f^J(p)$ are defined equivalently. They also contribute exponentially to their violation difference between data flow requirements and network characteristics but are presented as bars because their violations are time-invariant over multiple time slots. These components act as repelling forces.

Attracting forces pull non-allocated tokens to networks in general while repelling forces push allocated tokens away from non-matching time slots and networks.

Attracting forces arise, whenever tokens are not allocated, as desired. Accordingly, the first attracting force is non-allocated token cost $c_f^u(p^*)$. In addition, not allocating sufficient tokens within a defined time frame leads to minimum throughput violation $c_f^{tp}(p^*)$, which defines our final cost component of the model of forces.

$$c^{flow}(p) = \sum_{f \in F} \omega_f^{user} \cdot \left(\underbrace{c_f^{dl}(p) + c_f^{st}(p) + c_f^L(p) + c_f^J(p)}_{\text{repelling forces}} + \underbrace{c_f^u(p^*) + c_f^{tp}(p^*)}_{\text{attracting forces}} \right) \quad (4.6)$$

In the following sections, we present the mathematical modeling of the rating function. Therefore, we define the cost function components formally with equations and discuss them in detail.

4.1.2.3 *Token Allocation Model*

The token allocation model covers basic rules about how tokens can be allocated to networks. Allocated tokens create data traffic on networks, which in return creates monetary cost, acting as repelling force in the model. We assume that all monetary network cost models like pay-per-use or high-speed-volume flat rates can be linearized to a single input parameter as valid approximation over a short period. To calculate the cost function component from monetary cost, the parameter ω_n^{mon} weights allocated data tokens to each network n linearly. The abstract monetary cost $c^{mon}(p)$ of transmission plan p is further multiplied with the trade-off parameter willingness-to-pay, i.e. the monetary cost weight ω^{mon}, which balances monetary cost against the other components of the cost function, cf. Equation 4.7. Note that a large monetary cost weight ω^{mon} results in a low willingness-to-pay of the user.

Tokens of a flow, which are not allocated in a plan $u_f(p)$ lead to a requirement violation of the flow, cf. Equation 4.8. This violation handles general non-allocation and, thus, is modeled in the rating function as attracting force. To determine the number of non-allocated tokens, we subtract all planned tokens $p_{f,t,n}$ of a flow from the number of flow tokens to allocate $\widehat{p}_f$. To incorporate this violation from non-allocated data tokens as cost component into the rating function, we multiply the number of non-allocated tokens with a flow-specific weight ω_f^u. Hence, the violation contributes to the cost function with the cost component $c^u_f(p)$, according to Equation 4.9.

Allocation of tokens additionally underlies two constraints. Firstly, this is the capacity of the network. The maximum number of tokens, which a transmission plan p can assign to a network n in a time slot t, is limited by the network capacity in this time slot. This network capacity in time slot t is modeled as the token bucket size $B_{t,n}$. We define this first restriction as constraint C1 in Equation 4.10.

Secondly, to connect to a network, the mobile node needs an adequate network interface of type $i \in I$, like a WiFi transceiver or a mobile network modem. We define the number of interfaces of type i available for a mobile node using the parameter k_i. We present the according to constraint C2 in Equation 4.11. Thus, the mobile node is not able to connect to more networks of the same type than its number of available interfaces of that type. To model this constraint, we apply two operators. Firstly, the identity operator $\mathbb{I}(x = y)$, which returns 1 if x is equal to y, else 0. Secondly, we apply the signum function $\mathrm{sgn}(x)$, which returns -1 for negative values of x, 0 for $x = 0$ and 1 for positive values of x. In this case, it identifies if a network is used by any flow.

The token allocation model restricts the number of parallel usable networks to the number of client interfaces, limits data rates to the network capacity and models monetary cost for allocated and violation cost for non-allocated tokens.

$$c^{mon}(p) = \omega^{mon} \cdot \sum_{n \in N} \left(\omega^{mon}_n \cdot \sum_{\substack{f \in F, \\ t \in T}} p_{f,t,n} \right) \tag{4.7}$$

$$\forall f \in F: \quad u_f(p^*) = \widehat{p}_f - \sum_{\substack{t \in T, \\ n \in N}} p^*_{f,t,n} \tag{4.8}$$

$$\forall f \in F: \quad c^u_f(p^*) = \omega^u_f \cdot u_f(p^*) \cdot \omega^{user}_f \tag{4.9}$$

$$\mathrm{C1:} \ \forall t \in T, n \in N: \ \sum_{f \in F} p_{f,t,n} \leqslant B_{t,n} \tag{4.10}$$

$$\mathrm{C2:} \ \forall t \in T, i \in I: \ k_i \geqslant \sum_{n \in N} \left(\mathbb{I}(i_n = i) \cdot \mathrm{sgn}\left(\sum_{f \in F} p_{f,t,n} \right) \right) \tag{4.11}$$

The equations model the token allocation, including monetary network transmission cost for allocated tokens, violation cost for non-allocated tokens and the network restriction from the mobile node's available network interfaces.

4.1.2.4 *Throughput Model*

The throughput model defines the data rate of flows over time. It substantially differs from models currently used in state-of-the-art work in order to enable unified treatment of all data flows. It covers a novel throughput continuity requirement, which defines if a data flow must be transmitted continuously or *to which degree*

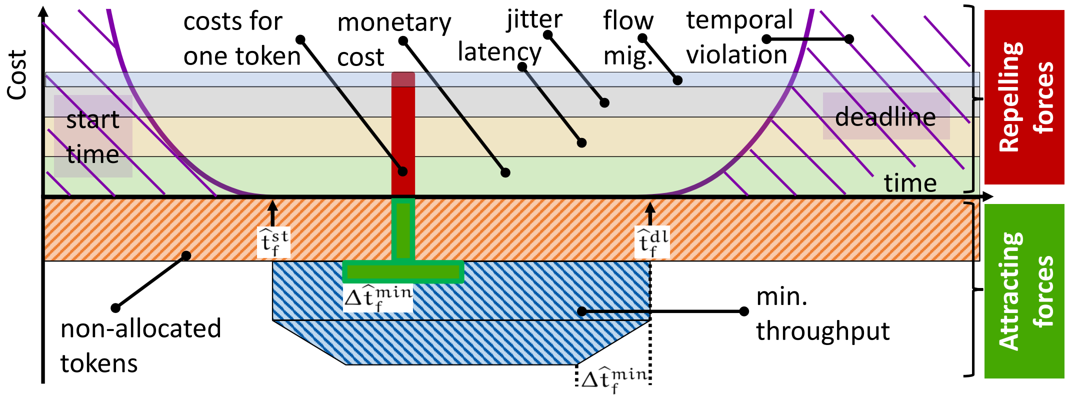


Figure 6: Cost visualization with the model of forces for one token allocation to a certain network

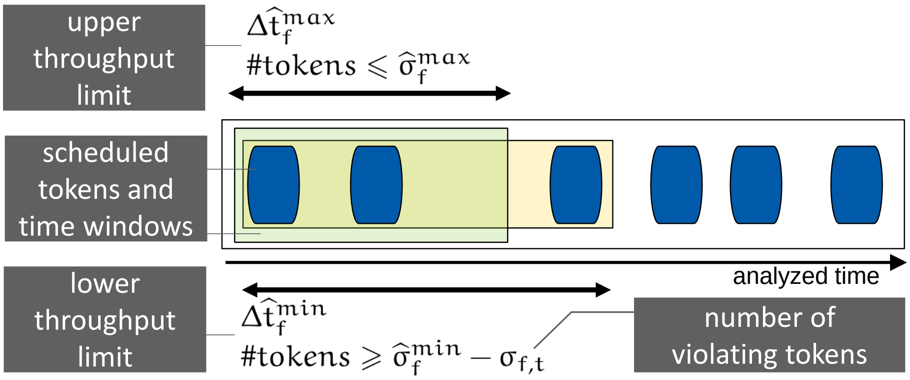

upper throughput limit
$\Delta\hat{t}_f^{max}$
$\#\text{tokens} \leqslant \hat{\sigma}_f^{max}$
scheduled tokens and time windows
analyzed time
lower throughput limit
$\Delta\hat{t}_f^{min}$
$\#\text{tokens} \geqslant \hat{\sigma}_f^{min} - \sigma_{f,t}$
number of violating tokens

Figure 7: Throughput model

it is allowed to be transmitted in bursts. This throughput continuity is an essential flow requirement, as it can cope with delay-tolerant as well as with delay-sensitive data flows. Treating all data flows in a unified way enables better optimization of data traffic planning, as follows. Reasonable transmission of delay-tolerant data in bursts is proven to increase transmission efficiency by Lu et al. [137], as the mobile node may reduce the load in bottleneck regions, releasing resources for non-delay tolerant data transmissions. However, the authors of that approach considered delay-tolerant data transmissions only and, thus, did not need a flow-specific flow continuity requirement. Models without such a flow continuity requirement either do not allow transmission in bursts at all. For example, classical network selection models, demanding for a fixed throughput in each time slot [59, 60, 39], do not intend transmission in bursts and, thus, waste the discovered optimization potential. This behavior represents the first extreme configuration of our throughput continuity requirement, demanding strict throughput continuity. In contrast, most transmission time selection approaches do not consider transmission continuity at all, defining the requirement as a simple deadline [15, 168, 133], which, in our model, corresponds to the maximum degree of allowing transmission in bursts and pauses anytime before the deadline. More sophisticated models head into the direction of this flow continuity, defining deadlines for subsets of data

Our novel throughput model covers the requirement, whether the flow's data transmission must be continuous or to which degree it may be in bursts. It enables unified assessment of delay-tolerant and delay-sensitive transmissions.

[137, 132], which might even be adapted dynamically to model acceptable start-up delays and pauses for media transmission, as shown by Antoniou and Stavrakakis [10]. In extension to that, our novel throughput model covers the configurable continuity requirement inherently, not requiring special treatment of corresponding data flows.

Our transmission rating function uses the basic definition of throughput for its requirement: an amount of data within a certain time span. Parametrization of the time span length allows definition whether the transmission must be instantly and continuous or to which degree it may be in bursts. The model covers two throughput limits, an upper and a lower one. The lower throughput limit corresponds to a classical QoS requirement, a minimum desired throughput. In our rating model, it represents an attracting force, fostering token allocation. It defines the minimum amount of data needed from the application during a certain time span to work properly. Network selection approaches fix this time span to one time slot, reducing throughput to a constant parameter during allocation. In contrast, our transmission rating model uses the parameter $\Delta\widehat{t}_f^{min}$ to define the time span as a window and an amount of tokens $\widehat{\sigma}_f^{min}$, which should be allocated in it. Note that the window size is defined relative to the selected time granularity $\Delta\tau$. Defining a larger window $\Delta\widehat{t}_f^{min}$, the tokens to be allocated can move freely in this time span and, hence, allowing the data transmission to occur in bursts. The larger the time window, the higher is the degree to which the transmission may happen in bursts. A window length of a single time slot, instead, enforces transmission of a data flow to be continuous and instant.

To show the effect of this model, we visualize an example in Figure 7. The outer frame represents the availability of a network over time to which tokens can be assigned. The blue shapes represent allocated tokens over time to this network while gaps in between them represent transmission pauses. For simplicity of the example, there are only a few data tokens, each representing an individual burst. The two smaller rectangles represent the throughput requirement windows. The upper throughput bound is painted in green with the shorter rectangle with length $\Delta\widehat{t}_f^{max}$, framing two tokens. The minimum throughput requirement is represented by the longer rectangle with length $\Delta\widehat{t}_f^{min}$, framing three tokens.

For evaluation of the transmission using the rating model, these rectangle windows slide over the analyzed period, from left to right, moving time slot wise between the start time and the deadline of the data flow. For each sliding state, the tokens inside the window are counted. To satisfy the upper throughput limit, the number of tokens in the window must never exceed the specified maximum $\widehat{\sigma}_f^{max}$, as modeled in Equation 4.12. This constraint C3 restricts the transmission plan to be realistic. It forbids to plan higher data rates than the application end point can deliver. In the example, a value of $\widehat{\sigma}_f^{max} = 3$ would comply the requirement. When sliding the green window from left to right, the number of tokens in the window never exceeds 3.

The upper throughput constraint limits the data rates to the maximum that applications can deliver. This is important for realistic planning over time.

The lower throughput requirement is modeled equivalently. However, the limit is softened with the additional component $\sigma_{f,t}$, as shown in Figure 7 bottom. This parameter models the number of tokens violating the minimum throughput limit in this sliding state of the window. Thus, the number of non-allocated tokens violating the requirement $\sigma_{f,t}$ are counted, as defined in Equation 4.13. Treating non-allocated tokens, the lower throughput model belongs to the attracting forces of our rating model and, thus, fosters token allocation. Assuming in the example a minimum throughput requirement of $\widehat{\sigma}_{f,t}^{min} = 4$, when the yellow colored rectangle, i.e. the minimum throughput window, slides from the left to the right, there exist are violations because of the transmission pause after the second burst. In the example, it takes effect from the first moment on until the yellow rectangle has passed the corresponding gap, reaching the end and covering the last four tokens. Accordingly, the model creates a force over the complete time span of a violation, meaning in contrary, that a single short high-throughput burst can satisfy the minimum throughput requirement as long as the window covers the burst. In Figure 6 shows the characteristic impacts of forces, this is marked by the green horizontal bar within the minimum throughput shape.

Note that only the time slots between start time $\widehat{t}_f^{st}$ and deadline $\widehat{t}_f^{dl}$ contribute to the cost function. It ensures that no token allocation outside those time limits is rewarded from the throughput model. In Figure 6, the throughput cost force is illustrated with a horizontal bar between the flow's start time and deadline extended with a trapezoid, resulting from the sliding window. Token allocation at a time slot t affects the throughput for all sliding window steps that cover the time slot, as exemplified above. Hence, the token allocation creates a reward for close-by time slots within the range of one time window. For the most left sliding window step, at the start time of the flow, the green bar in Figure 6 is cut off because only an overlap with the blue shape leads to a reward. Hence, the possible reward for token allocation is smaller near the time limits start time and deadline. This characteristic is intended and leads to two inherent advantages of the model. Firstly, throughput violation close to the start time is rated less severe than in the center, which might result in allocation of fewer tokens in this area. Indeed, common flow control algorithms like from TCP ramp up the throughput of a data flow. Therefore, the underrating in these time slots finally leads to a more realistic data flow planning. A similar effect holds for the deadline. The underrating in those time slots may lead to less allocated data traffic close to the deadline and potentially shifts data transmissions to earlier time slots, leading to a more conservative transmission near deadlines and keeping transmission plans better executable in reality.

Finally, rewarding neighboring tokens leads to an imbalance between flows. The number of counted tokens depends linear on the window size $\Delta\widehat{t}_f^{min}$ and the number of tokens to be allocated $\widehat{\sigma}_f^{min}$. This linear dependence could overrate high throughput data flows, which are allowed to be in strong burst, e.g. video on demand. To create fairness between the data flows, we normalize the resulting cost $c_f^{tp}(p^*)$ to them. Additionally, to counteract general underestimation of the throughput requirement through normalization, we scale it with parameter γ. Finally, the result is weighted by ω_f^{tp} to return the throughput violation cost of a flow, specified in Equation 4.14.

In contrast to the other cost components, minimum throughput cost is limited. As soon as the required minimum throughput is reached, this attracting force loses its effect. Thus, each token allocation reduces the remaining cost saving potential.

From the schematic view in Figure 6, each token allocation reduces the blue shape by the red bar. This reduction constitutes the stateful characteristic of minimum throughput requirement. Hence, applying the token allocation from the current transmission plan p reduces the attracting forces.

$$\text{C3:}\ \forall f \in F, t_0 \in T: \quad \hat{\sigma}_f^{max} \geqslant \sum_{t=t_0}^{t_0+\Delta\hat{t}_f^{max}} \sum_{n \in N} p_{f,t,n} \tag{4.12}$$

$$\forall f \in F, t_0 \in T: \quad \sigma_{f,t}^{min}(p) = \max\left(0,\ \hat{\sigma}_f^{min} - \sum_{t=t_0}^{t_0+\Delta\hat{t}_f^{min}} \sum_{n \in N} p_{f,t,n}\right) \tag{4.13}$$

$$\forall f \in F: \quad c_f^{tp}(p) = \frac{\gamma \cdot \omega_f^{tp}}{\Delta\hat{t}_f^{min} \cdot \hat{\sigma}_f^{min}} \cdot \sum_{t=\hat{t}_f^{st}}^{\hat{t}_f^{dl}-\Delta\hat{t}_f^{min}} \sigma_{f,t}^{min}(p) \tag{4.14}$$

4.1.2.5 *Deadline, Start Time, Latency and Jitter Models*

Squared derived violation weights make many small violations less severe than few strong violations.

The models for the deadline, start time, latency and jitter requirements, shown in Equations 4.15 to 4.18, are very similar to each other. All of them are based on a difference of two values expressing a requirement violation. To model the cost component, we square violation differences and weight them linearly. Squaring of violations leads to an important effect: It rates heavy violations more severe than many small or even negligible violations. It fosters transmission planners to avoid strong violations whenever possible.

For deadline violation, we determine the time by which the currently analyzed time slot exceeds the deadline. The start time is modeled equivalently. Both relate to absolute time. Hence, their equations contain the time granularity $\Delta\tau$. Latency and jitter violations are both modeled using abstract values. Their violation is squared and weighted. Moreover, we argue that time violations as well as latency and jitter violations are perceived by the user over time, but it is not relevant how much data is transmitted in that time slot. To model this characteristic, we use the signum function for all of them and, thus, restrict violation to take effect just once for a time slot and not separately for each allocated token. Accordingly, the four cost components belong to the repelling forces of our rating model, which push data tokens away from non-matching time slots and networks.

$$\forall f \in F: \quad c_f^{dl}(p) = \omega_f^{dl} \cdot \sum_{\substack{t \in T \\ n \in N}} \operatorname{sgn}(p_{f,t,n}) \cdot \max(0, \Delta\tau \cdot (t - \hat{t}_f^{dl}))^2 \tag{4.15}$$

$$\forall f \in F: \quad c_f^{st}(p) = \omega_f^{st} \cdot \sum_{\substack{t \in T \\ n \in N}} \operatorname{sgn}(p_{f,t,n}) \cdot \max(0, \Delta\tau \cdot (\hat{t}_f^{st} - t))^2 \tag{4.16}$$

$$\forall f \in F: \quad c_f^L(p) = \omega_f^L \cdot \sum_{\substack{t \in T \\ n \in N}} \operatorname{sgn}(p_{f,t,n}) \cdot \max(0, L_n - \widehat{L}_f)^2 \tag{4.17}$$

$$\forall f \in F: \quad c_f^J(p) = \omega_f^J \cdot \sum_{\substack{t \in T \\ n \in N}} \operatorname{sgn}(p_{f,t,n}) \cdot \max(0, J_n - \widehat{J}_f)^2 \tag{4.18}$$

4.1.2.6 *Network Association and Flow Migration Model*

A transmission plan can switch the networks over time, which are used for transmission of a data flow. This is called a handover. Handovers cause some protocol signaling overhead and often lead to a performance degradation of the transmission during the process [7, 38, 129]. Therefore, they should be applied only when necessary. For this reason, we integrate a component for them into the cost function.

Handover identification for a data flow in a given transmission plan is not trivial to model mathematically because transmission may pause and the handover destination network is unknown. Therefore, a handover may span all networks and all succeeding time slots in the planning horizon. To identify these events using a mathematical model, we define network associations: Each data flow has an association to exactly one network in each time slot, cf. Equation 4.19. An association signifies a possible transmission but does not imply it for the current time slot, cf. Equation 4.20. Finally, a change of the network requires a change in network association. We call this change of network association a flow migration. It specifies the order and signaling action to migrate a data flow from one network to another, independent of its actual data transmission. Since network association exists for each time slot, the mathematical model can be simplified to check for changes in consecutive time slots. Equation 4.21 achieves this. It checks for differences in consecutive time slots. The cost component c^{mig} covers the flow migration weight ω^{mig} and, in addition, the component $1/2$ because the component identifies each flow migration twice, once for the origin network and once for the target network. This way, flow migrations can be identified analyzing network associations.

$$\forall f \in F, t \in T: \quad \sum_{n \in N} a_{f,t,n} = 1 \tag{4.19}$$

$$\forall f \in F, t \in T, n \in N: \quad a_{f,t,n} \geqslant \operatorname{sgn}(p_{f,t,n}) \tag{4.20}$$

$$c^{mig}(p) = \frac{\omega^{mig}}{2} \cdot \sum_{\substack{f \in F \\ n \in N}} \sum_{\substack{t \in T | \\ t \geqslant 2}} (a_{f,t-1,n} - a_{f,t,n})^2 \tag{4.21}$$

Table 2: Overview of rating model parameters and variables

Symbol	Description
$f \in F$	flow in flows to be allocated
$t \in T$	time slot in overall planned time slots (time horizon)
$n \in N$	network in available networks (in time horizon)
$i \in I$	network interface type i in interface types I
$\Delta\tau$, τ	time slot duration, time τ
d, d_f, $d_{t,n}$	data token size, data of flow, capacity of network in time slot t
p	a transmission plan, consisting of token allocations
$p^*(=p)$	indication that the employed function accounts for non-allocated tokens
$\widehat{p}_f$	number of tokens of flow f to be allocated
$B_{t,n}$	capacity of a network n at time slot t (bucket size)
$p_{f,t,n}$	number of allocated tokens of flow f to network n at time slot t
$a_{f,t,n}$	flow association $\in \{0,1\}$ of flow f in time slot t to network n
k_i	number of available network interfaces of technology i
$\Delta\widehat{t}_f^{max}$	size of max. throughput window of flow f in time slots
$\widehat{\sigma}_f^{max}$	max. amount of tokens in max. throughput window of flow f
$\Delta\widehat{t}_f^{min}$	size of min. throughput window of flow f in time slots
$\widehat{\sigma}_f^{min}$	min. amount of tokens in min. throughput window of flow f
$\sigma_{f,t}$	violation strength in tokens of min. throughput requirement
$\widehat{t}_f^{dl}$	deadline requirement $\in T$ of flow f
$\widehat{t}_f^{st}$	start time requirement $\in T$ of flow f
L_n, $\widehat{L}_f$	latency of network n, latency requirement of flow f
J_n, $\widehat{J}_f$	jitter of network n, jitter requirement of flow f
$\omega_{(f)}^{x}$	weight for parameter x (of flow f)
$c(p)$	total cost of plan p
$c^{attr}(p^*)$	cost from attracting forces in plan p
$c_f^{u}(p*)$	cost for non-allocated (unscheduled) tokens of flow f in plan p
$c_f^{tp}(p*)$	cost for minimum throughput requirement violations of flow f in plan p
$c^{rep}(p)$	cost from repelling forces in plan p
$c^{mon}(p)$	cost from monetary cost of network ω_n^{mon} in plan p
$c^{mig}(p)$	cost from flow migration (handover) in plan p
$c_f^{dl}(p)$	cost from deadline violation of flow fs of flow f in plan p
$c_f^{st}(p)$	cost from start time violation of flow f in plan p
$c_f^{L}(p)$	cost from latency violation of flow f in plan p
$c_f^{J}(p)$	cost from jitter violation of flow f in plan p

These equations finalize the novel data transmission rating model. It covers the three objectives of (1) application QoS requirement violation, (2) monetary cost and (3) flow migration. These three objectives are linearly combined to form the unified cost function c. It covers essential elements of network selection and extends them with temporal models from time selection, i.e. our novel throughput model, start times and deadlines. Thus, it defines a joint rating for time-network selection in data transmission. An overview about employed variables and parameters is shown in Table 2.

Minimizing the defined cost function $c(p)$ subject to the constraints C1 to C3 defines the transmission planning problem. Due to the rating model, validity and cost of a transmission plan p can be determined in polynomial time. In addition, the knapsack problem [143], which is known to be NP-complete, defines a sub-problem of the transmission planning problem, showing that the transmission planning problem is not in NP. There is no algorithm known to solve it in polynomial time. Thus, we argue that the transmission planning problem is NP-hard. We detail this argumentation in Appendix A.1.

To target this issue, we develop transmission planning algorithms that use heuristics to create good transmission plans in polynomial time. Their strategies and designed heuristics are detailed in the following section.

4.2 JOINT TIME-NETWORK SELECTION IN TRANSMISSION PLANNING

We design JTP, a joint time-network selection strategy, and compare it to leading strategies from the state-of-the-art.

In the previous section, we presented the rating model for transmission plans, which enables evaluation of joint time-network selecting transmission strategies. These transmission strategies form the focus of this section. As a main contribution, we design the strategy Joint Transmission Planning (JTP) that explicitly applies a joint time-network selection. For evaluation purpose, we adopt three transmission strategies reflecting predominant state-of-the-art concepts. The first strategy is a classical Network Selection (NS), as found in related work and detailed in Section 3.1. Second, we present an Opportunistic Network Selection (ONS), which extends NS by the opportunity to omit data transmission if the transmission is not considered beneficial enough. Even though we did not find a direct correspondence to ONS in related work, we consider it as an upper performance bound for network selection approaches, combining different concepts, slightly extended by concepts from time selection approaches. As third strategy we compare a classical Delayed WiFi Offloading (DWO), as detailed in state-of-the-art approaches detailled in Section 3.2, that prefers to send data via WiFi, planning ahead delay-tolerant data transmissions without considering detailed network characteristics. In the course of this, we investigate hypothesis H1:

H1: Transmission benefits significantly from joint time-network selection.

Current approaches usually consider only one of those dimensions. H1 covers the idea that rating function minima are spread in this solution space and, hence, cannot be exploited with focusing on only one of the two dimensions. State-of-the-art network selection (NS) methods cover only data distribution over present-available networks [55, 207, 86, 138]. In contrast, delayed offloading [145, 131, 40] and resource allocation [26] schemes do not consider the network selection sufficiently.

They either move data transfers only temporally or simplify network selection to, e.g., a WiFi-preferred strategy. Both concepts show essential benefits on their own. H1 emphasizes the idea, that a combination of the two concepts, a joint time-network selection, can unlock a hidden optimization potential and outperform existing approaches.

4.2.1 *Transmission Planners with Different Time-Impact*

To investigate the above-mentioned hypothesis H1, we develop a novel time-network selection strategy, called Joint Transmission Planning (JTP), and the three approaches, which differ either in their transmission time selection or network selection strategy. Note that we assume complete knowledge about the future network connectivity and data to send. Hence, the presented strategies conduct a transmission planning with perfect prediction. The problem of erroneous prediction is handled in Chapter 5.

For comparability of the approaches, the four transmission strategies have one thing in common: they employ the same heuristics as sub-functions, which we derive in the following sections from the transmission rating model, presented in Section 4.1. The first heuristic method determines a data flow priority and sort data flows accordingly SORTFLOWS(FLOWS). The second one is called SORTNETMATCH(FLOW, NETWORKS) and estimates each network match for a data flow, finally sorting the networks accordingly. Finally, the third heuristic method estimates if a token allocation is beneficial, representing an opportunistic transmission decision GETMATCHBENEFIT(FLOW, TIME, NETWORK, PLAN). In addition, we introduce a method ALLOCATETOKENS(FLOW, NETWORK, PLAN), which allocates tokens of a flow to a network, if the transmission satisfies the constraints C1-C3. It guarantees that resulting transmissions and plans are feasible. Note that the four transmission strategies are realized as a depth-first search, finding a solution fast, with search-ordering and forward-checking, according to Section 2.4. Hence, they do not explore different opportunities in each search step but follow the one that appears to be best. Search-ordering is covered by the strategies themselves, as well as by the heuristics for benefit calculation, flow and network prioritization, controlling the search path to take. Forward-checking is used to avoid transmissions from violating the constraints, implemented in the token allocation method. As the goal of the analysis in this chapter is comparison of joint time-network selection to pure network and pure transmission time selection, the application of further optimization methods to the presented strategies is out of scope. The named heuristics are used equally from all three transmission planners to enable a fair comparison of their search strategies. In the following, we present the transmission planning strategies and, subsequently, detail the above-mentioned heuristic sub-functions used by the strategies.

The derived heuristics are employed in all approaches equally to enable a fair comparison of the underlying transmission strategies.

4.2.1.1 *Network Selection and Opportunistic Network Selection*

Our first strategy is a pure Network Selection (NS), representing the most advanced algorithms of this class, as discussed in Section 3.1. It does not cover time selection. Accordingly, Network Selection does only consider present-available networks and ignores any information about the future. Furthermore, following the most advanced approaches in this field, it selects networks individually for each data flow and employs multi-homing to enable concurrent transmissions via different networks. To make NS represent an upper bound performance for typical network selection approaches, we eliminate the impact of incomplete information, assuming NS to know the exact characteristics of each present-available network. It prioritizes data to transmit, using the sub-function sortFlows(flows), and assigns the prioritized data to the best matching present-available networks, identified using the sub-function sortNetMatch(flow, networks), always transmitting in best-effort fashion, i.e. as much as possible.

Opportunistic Network Selection introduces the opportunity to delay data transmission statistically whenever the flow-network match is considered to be insufficient.

The second strategy is an Opportunistic Network Selection (ONS), which is based on Network Selection and extends it by an opportunistic component from time selection, deciding whether to transmit or not. Thus, it employs the token assignment decision sub-function assignmentDecision(flow, network, plan), as detailed in Section 4.2.2, estimating the benefit of the transmission opportunity and comparing it to a threshold c_{lim}, which is set to 0 in the default case. If the estimated benefit is lower than the threshold c_{lim}, then the network is considered as insufficient and no transmission is triggered, assuming that there will be a better matching transmission opportunity in the future. This behavior results in statistical delaying of data transmissions, integrating the time dimension into network selection statistically. A similar time selection approach was presented by Balasubramanian et al. [15], however without employing sophisticated network selection algorithms. Even though we did not find any directly corresponding transmission strategy in related works, we consider ONS as upper transmission performance bound for early adopters of time-network selection, representing a statistical online network selection with a limited time selection. Note that both strategies, NS and ONS, are online approaches, considering only the present environment and data to transmit.

Their procedure is given as pseudocode in Algorithm 1. It starts in line 2 with the creation of an empty transmission plan to initialize the output variable. Next, the algorithm sorts data flows according to their priority, using the heuristic sub-function sortFlows(flows). Since Network Selection and Opportunistic Network Selection work on time slots consecutively, they can be applied as online algorithms. For each time slot, they iterate over the data flows, which are sorted according to their priority. In these iterations, the sub-function getMatchBenefit(flow, time, network, plan) sorts networks according to their flow-network matching to the current data flow to assign. Finally, the algorithm comes to the point in which the strategies differ: token assignment decision. Network Selection assigns data whenever possible. In contrast, Opportunistic Network Selection uses the method assignmentDecision(flow, network, plan) in line 9 to decide whether to assign tokens of the current data flow or not.

4.2.1.2 *Delayed Wifi Offloading*

The third strategy, Delayed WiFi Offloading (DWO), represents most advanced transmission time section approaches, e.g. of Cheung and Mehmeti [41, 145]. The strategy DWO plans data transmission ahead with the preference to use prospec-

Algorithm 1 Network Selection (NS) and Opportunistic Network Selection (ONS)

```
procedure NetworkSelectionBase(flows, networks)
    plan ← empty plan
    flows ← sortFlows(flows)                          ▷ sort data flows by priority
    for time in Time do                               ▷ progress of time
        for flow in flows do
            ▷ for current time slot, assign to best matching available network
            networks ← sortNetMatch(flow, networks)
            for network in networks do
                if assignmentDecision(flow, network, plan) then
                    ▷ assign tokens to network, satisfying constraints
                    plan ← assignTokens(flow, time, network, plan)

                                                      ▷ Network Selection (NS)
procedure assignmentDecision(flow, time, network, plan)
    return true

                                      ▷ Opportunistic Network Selection (ONS)
procedure assignmentDecision(flow, time, network, plan)
    return getMatchBenefit(flow, time, network, plan)< $c_{lim}$
```

tive WiFi resources but uses mobile networks for all data which cannot be offloaded to WiFi without violating the data flow's deadline. However, they do not consider selecting between more than one mobile network or different WiFis and, thus, reduce the network selection to WiFi-preferred, not distinguishing between different mobile or WiFi networks. To eliminate any performance impact from incomplete information and let the strategy DWO represent an upper bound performance approach for time selection, we assume present-available and future-available networks as well as future data flows to be known a priori. Hence, DWO sorts data flows, using the sub-function sortFlows(flows), and sorts prospective available networks by their type, i.e. WiFi first. Subsequently, it plans ahead allocation of their data to the WiFi networks, which will be available within the data flow's deadline, and allocates the remaining data to free resources of other networks. This represents an explicit transmission time selection, according to the prospective availability of WiFi networks.

Delayed WiFi Offloading plans transmissions over a time horizon, favoring WiFi networks without distinguishing transmissions with respect to the flow-network match.

Its procedure is presented in Algorithm 2 and implements the time selection according to network resources availability and time constraints. For initialization, lines 2-4 create an empty plan and sort networks by type, resulting in a WiFi-first order, and data flows according to their priority, using the heuristic sub-function sortFlows(flows). For each data flow in the sorted list, it iterates over the type-ordered networks. Within the desired transmission period of the data flow, DWO tries to assign tokens to the networks, using the allocateTokens(flow, network, plan) function, which ensures constraint satisfaction of C1-C3. Accordingly, an assignment only proceeds if, firstly, the network has free resources left in the desired transmission time frame of the data flow, secondly, the data flow may send additional data in the treated time slot and, thirdly, the network interface may currently access the desired network. The planning characteristic of DWO, in contrast to the online characteristic of NS and ONS, is reflected in moving the time selection loop into the network selection loop. This implies temporal independence of the network selection from time, achieved through selecting on all networks predicted to be available within the planning time horizon.

Algorithm 2 Delayed WiFi Offloading (DWO)

```
procedure DelayedWiFiOffloading(flows, networks)
    plan ← empty plan
    networks ← sortWiFiFirst(networks)    ▷ sort networks by type: WiFi first
    flows ← sortFlows(flows)              ▷ sort data flows by priority
    for flow in flows do
        for network in networks do
            for time in flow.startTime to flow.deadline do    ▷ time selection
                                ▷ assign tokens to network, satisfying constraints
                plan ← assignTokens(flow, time, network, plan)
```

4.2.1.3 *Joint Transmission Planning*

Our fourth strategy is Joint Transmission Planning (JTP), covering our contribution of an explicit time-network selection. Like DWO and in contrast to the two previous strategies NS and ONS, Joint Transmission Planning considers all networks available in the complete time horizon to plan for selection. As for the other approaches, we assume complete knowledge about data flows and networks to be available.

Joint Transmission Planning covers an explicit time selection, which relies on the availability of matching networks.

Joint Transmission Planning focuses on the flow-network matching, employing the heuristic sub-function sortNetMatch(flow, networks) and treats the desired transmission time limit satisfaction of the data flow as a constraint for data allocation. Accordingly, the algorithm selects those time slots for the data token allocation, in which the best matching networks are, respectively will be, available. Considering future-available networks and their characteristics for transmission planning, Joint Transmission Planning explicitly integrates time dimension into its selection strategy.

The procedure of JTP is shown in Algorithm 3. After initialization of an empty plan and sorting of data flows in lines 2-3, using sortFlows(flows), for each data flow, the dedicated flow-network matching is calculated and networks are ordered accordingly in lines 4-6, using sortNetMatch(flow, networks). For Joint Transmission Planning, as for Delayed WiFi Offloading, the time selection loop is subordinate to the network selection loop. Hence, it focuses on flow-network matching, treating desired transmission time limits as a constraint, as shown in lines 8-9. Joint Transmission Planning additionally reuses the transmission decision mechanism getMatchBenefit(flow, time, network, plan) of ONS, as shown in line 10, comparing an estimated benefit of the transmission to a threshold c_{lim}, which enables opportunistic delaying of data transfers beyond the planning time horizon.

Further constraint satisfaction is ensured in line 12, using the function allocateTokens(flow, network, plan).

In a real system, this results in an entirely different algorithm design. The realization of the two presented online network selection algorithms NS and ONS, indeed, do not cover the time loop at all. In fact, the top-level time selection loop from Algorithm 1 represents time progress in reality. In contrast, Joint Transmission Planning introduces the time selection as a new dimension in its algorithm, enclosing the time selection explicitly and moving it inside the network selection in order to select between all networks in the time horizon. This enlarges the algorithm's solution space, unlocking a new optimization potential by integrating explicit time selection into dedicated network selection.

Algorithm 3 Joint Transmission Planning (JTP)

```
 1: procedure JointTransmissionPlanning(flows, networks)
 2:     plan ← empty plan
 3:     flows ← sortFlows(flows)                       ▷ sort data flows by priority
 4:     for flow in flows do
 5:                                    ▷ sort networks according to their match to flow
 6:         networks ← sortNetMatch(flow, networks)
 7:         ▷ for best matching network, assign to constraint satisfying time slots
 8:         for network in networks do
 9:             for time in flow.startTime to flow.deadline do      ▷ time selection
10:                 if assignmentDecision(flow, time, network, plan) then
11:                                   ▷ assign tokens to network, satisfying constraints
12:                     plan ← assignTokens(flow, time, network, plan)
13:
14:                                           ▷ Joint Transmission Planning (JTP)
15: procedure assignmentDecision(flow, time, network, plan)
16:     return getMatchBenefit(flow, time, network, plan)< $c_{lim}$
```

4.2.2 *Heuristics*

The three transmission planners use common heuristics to make crucial decisions. Using the same heuristics for each algorithm makes the strategies comparable and proves the benefits of the conceptual algorithm design. The first one is the sortFlows(flows) heuristic, the second is the sortNetMatch(flow,networks) heuristic. Joint Transmission Planning and Opportunitic Network Selection additionally use the heuristic getMatchBenefit(flow,time,network,plan).

sortflows(flows) An essential part of the algorithms is flow prioritization. Whenever the network resources are not sufficient, flow prioritization decides which data to assign and which data to drop. In our algorithms, we assign tokens of one flow after another. Hence, priorities resemble an allocation order. Therefore, our heuristic calculates a restrictiveness value $r(f, p_0)$. We calculate restrictiveness according to Equation 4.22, which we derive from components of the rating function.

The restrictiveness heuristic estimates the priority of a data flow for transmission.

$$r(f, p_0) = \omega_f^{user} \frac{c_f^u(p_0) + c_f^{tp}(p_0)}{\widehat{p}_f} \tag{4.22}$$

This order is determined from the method sortFlows(flows). The restrictiveness value depends on the cost that arises when the flow is not allocated as desired, hence, using the empty transmission plan p_0 for $c^{flow}(p_0)$. According to our model of forces, only attracting forces exist when no tokens are allocated. Hence, we use only the components of the attracting forces for the restrictiveness heuristic. This resembles the non-allocated tokens $c_f^u(p_0)$ and the minimum throughput cost $c_f^{tp}(p_0)$, which we weight with the user preference ω_f^{user}. Moreover, to receive a metric independent from the data amount, we normalize the cost to the number of tokens $\widehat{p}_f$ of the flow. This normalization creates fairness between huge data flows and tiny ones.

Why do the repelling forces not contribute to the restrictiveness of a data flow? Repelling forces usually represent violations. Hence, it depends on the network to allocate tokens on whether repelling forces have an effect or not. Depending on the scenario, incorporating the repelling forces into the restrictiveness metric might have positive effects or even negative effects on flow prioritization. For example, when there are only networks available satisfying all application QoS requirements, most repelling forces have no effect. However, the impact cannot be estimated without a detailed analysis of the expected network environment. Since we want the heuristic to be applicable for our online strategies that do not have detailed knowledge about the future, an estimation about the effect of repelling forces is not applicable. Hence, we do not use repelling forces in the restrictiveness heuristic.

The forces are illustrated in Figure 6 on page 36. The non-allocated token cost $c_f^u(p_0)$, shown as orange-striped bar along the entire time horizon, is independent of time. It is linear to the amount of data of the flow. In contrast, the minimum throughput violation cost is restricted in time to the period between the flow's start time $\widehat{t}_f^{st}$ and its deadline $\widehat{t}_f^{dl}$, illustrated as a blue striped trapezoid.

The flexibility in time dimension plays a crucial role in transmission planning. Flow tokens whose transmission is delay-tolerant can tentatively be assigned to different points in time as long as deadlines are met. This flexibility in the time dimension reduces their restrictiveness value and, hence, should also decrease their priority to be allocated. The window parameter $\Delta\widehat{t}_f^{min}$ from or throughput requirement model in Section 4.1.2.4 reflects this flexibility and seems to be predestined for this purpose. However, it is not a mandatory parameter for data flows requirement definition and, thus, it does not exist for all flows. Therefore, the throughput window does not enable a fair comparison between all flows. However, to derive an estimation, we select the time difference between the start time and the deadline. It does not reflect the flexibility but, however, correlates with it. Thus, the time difference between start time and deadline limits the maximum flexibility of data flows for movement in time dimension. Accordingly, the time difference between start time and deadline is the supremum of the throughput window length, i.e. its least upper bound.

However, the difference between start time and deadline can reach large values. Furthermore, as mentioned above, the flexibility of the data flows is only tentative. When used for normalization of the restrictiveness, the difference influences the result too heavily to reflect a tentative trend. As a solution, we apply the logarithm to consider just its magnitude and use it in the denominator of the restrictiveness. Furthermore, the time difference between start time and deadline measures at least one time slot. Hence, the logarithm can become zero. To make the heuristic applicable even for this case of short time differences, we add 1, which avoids the denominator of the restrictiveness equation to become zero. Accordingly, the equation of the restrictiveness including temporal flexibility of data flows $r^{flex}(f, p_0)$ is given in Equation 4.23. Finally, the SORTFLOWS(flows) method sorts the data flows according to their restrictiveness $r^{flex}(f, p_0)$.

$$r^{flex}(f, p_0) = \omega_f^{user} \frac{r(f, p_0)}{1 + \log(t_f^{dl} - t_f^{st})} \qquad (4.23)$$

SORTNETMATCH(FLOW,NETWORKS) A major problem of transmission planning is to find out, how good a token assignment of a flow matches to networks: network selection. Equivalently to the previous method for flow prioritization, we create a sorted list, here with preferred networks for each data flow. In this heuristic, the repelling forces of our rating model play the major role. Hence, we take violations of latency and jitter into account. In addition, we consider the monetary cost of the networks. While latency and jitter rely on the flow-network match, the monetary cost depends on the selected network only. Nevertheless, they belong to the repelling forces and are independent of time, as visualized in Figure 6. To comply with the design of the rating function, we use a satisfactory metric. Hence, we calculate the latency and jitter matches, according to Equation 4.24. Like the rating function, it covers the satisfactory quadratic mismatch and flow-specific weights of the potential violations. However, there is an ambiguity between the heuristic and the rating function design: While latency and jitter both rely on a single time slot in the rating function, the heuristic estimates the match normalized to a data token. Hence, we have to normalize these two terms by the expected number of tokens in a time slot. For this estimation, we use the average minimum throughput. Therefore, we divide the minimum throughput model's token amount $\widehat{o}_f^{min}$ by the dedicated time window length $\Delta\widehat{t}_f^{min}$. In many cases, the algorithm will be able to allocate more tokens than the minimum to the network. Hence, the heuristic tendentiously overestimates the effect of the requirement violation. However, a better estimation for token allocation is hard to get. A second option is to use the maximum throughput limit, which, in contrast, tendentiously underestimates the effect. We prefer the overestimation of the effect over an underestimation since it represents the more conservative option and usually gets closer to the actual result. Nevertheless, a requirement satisfaction is still always recognized correctly. If the

A data flow's network match is estimated using a heuristic function of the repelling forces of the rating model.

minimum throughput requirement is not defined for the flow, we use the average throughput of the network instead, as modeled in Equation 4.25.

$$v_{net}(f,n) = \omega_f^{user} \cdot \frac{(\omega_f^L \cdot \max(0, L_n - \widehat{L}_f)^2 + \omega_f^J \cdot \max(0, J_n - \widehat{J}_f)^2}{\sigma_f^{exp}} + \omega^{mon} \cdot \omega_n^{mon} \tag{4.24}$$

$$\sigma_f^{exp} = \begin{cases} \frac{\widehat{\sigma}_f^{min}}{\Delta \widehat{t}_f^{min}} & \text{if } \widehat{\sigma}_f^{min} \cdot \Delta \widehat{t}_f^{min} > 0 \\ S_n^{average} & else \end{cases} \tag{4.25}$$

Finally, the sortNetworksMatch method uses the heuristic $v_{net}(f,n)$ to sort the networks according to their match to a flow f. On a tie, we prefer the network with higher capacity, potentially allowing more additional data flows to be allocated on this network. This happens especially within network environments that meet all application QoS requirements. Conclusively, $v_{net}(f,n)$ provides a heuristic network preference for transmission planning. However, the complete network selection is accomplished within the next presented function.

getMatchBenefit(flow,time,network,plan) An important characteristic, balancing the attracting and repelling forces of a flow allocation, is the getMatchBenefit function. It determines how much benefit the allocation of one token of flow f to network n at time slot t achieves. Therefore, we focus on the action of allocating one token of a flow to a certain network at a certain time slot, which finally resembles network selection. We isolate the forces caused by this allocation and subtract the repelling from the attracting forces. If the attracting forces dominate, the heuristic claims that the analyzed allocation is beneficial. The function approximates essential parts of the rating function, normalized to a single token.

To estimate the benefit of a data token allocation, we rely on the difference of heuristically estimated attracting and repelling forces of the rating model.

To estimate the repelling forces $c^{\widetilde{rep}}(f,t,n)$ of a specific allocation with respect to latency, jitter and monetary cost, we reuse the violation function $v_{net}(f,n)$ from sortNetMatch. Furthermore, we neglect the flow migration cost in the heuristic because it creates a stateful temporal component, which is hard to handle but has no significant effect on the final result. In addition, we consider time limits, namely start time and deadline. Since these forces appear, like latency and jitter, per time slot and not per token, we define their estimation equivalently. As presented in Equation 4.25, we use the sum and normalize it by the expected number of tokens in this time slot σ_f^{exp}. Equivalently to the network violation function, we follow the rating function design and use the satisfactory quadratic mismatch, balanced by the flow-specific weights. The result for the time limit violation heuristic $v_{time}(f,t)$ is shown in Equation 4.26.

$$v_{time}(f,t) = \omega_f^{user} \cdot \frac{\omega_f^{st} \cdot \max(0, \widehat{t}_f^{st} - t)^2 + \omega_f^{dl} \cdot \max(0, t - \widehat{t}_f^{dl})^2}{\sigma_f^{exp}} \tag{4.26}$$

To estimate the attracting forces $c^{\widetilde{attr}}(f,p)$ of non-allocated tokens of a flow, we reuse the model from the restrictiveness metric $r(f,p)$. However, we use the actual transmission plan p for the rating instead of the empty plan p_0. The essential difference from using the actual plan resides from the minimum throughput requirement model, detailed in Section 4.1.2.4, because a token allocation creates only a benefit as long as the minimum throughput is not reached.

$$\textsc{getMatchBenefit}(f,t,n,p) = \underbrace{r(f,p)}_{\widetilde{c^{attr}}(f,p)} \underbrace{-v_{net}(f,n) + v_{time}(f,t)}_{\widetilde{c^{rep}}(f,t,n)} \tag{4.27}$$

$$\textsc{assignmentDecision}(f,t,n,p) : \textsc{getMatchBenefit}(f,t,n,p) > c_{lim} \tag{4.28}$$

The presented heuristics estimates the forces. These forces are summed up in Equation 4.27. To identify the balance, it subtracts the attracting forces from the repelling forces. Assuming the heuristics to estimate the forces accurately, the farther below zero the result of this heuristic is, the more cost can be saved with a token allocation. According to Equation 4.28, the parameter c_{lim} can be used to cure the heuristic's allocation imbalance or to allocate with a biased threshold. A positive value signalizes that it might be beneficial to assign the currently analyzed tokens to this time slot and network. Keep in mind, that Opportunistic Network Selection and Joint Transmission Planning use this metric to decide whether a token is allocated or not. With a limit constant of $c_{lim} = 0$, a negative sign shows domination of the attracting forces and, hence, fosters a token allocation according to the algorithms. In contrast, a negative sign indicates that allocation at this network and time slot is potentially not beneficial and should be avoided. Note that all three transmission planning algorithms use the three presented heuristics. Therefore, they differ only in their structure. This similarity enables a fair comparison of their underlying strategies in the evaluation.

4.3 EVALUATION

To evaluate our developed transmission planners, we present the evaluation design and analyze the results in this section. Firstly, we introduce the dependent variables and evaluation metrics. Consecutively, we present the simulation setup with the independent and controlled variables. Finally, we describe and discuss the evaluation results.

4.3.1 *Evaluation Metrics and Dependent Variables*

Transmission plans p are defined by a number of allocated tokens $p_{f,t,n}$, which belong to a flow f and are allocated at a time slot t to a network n. Allocated tokens define the primary decision variable of the problem. To rate the quality of such transmission plans, we introduced the rating function in Section 4.1. It provides a quality metric for data transmission plans p and rates application QoS requirement satisfaction as well as monetary cost. This cost function is designed for performance comparison between different transmission plans for the same scenario. Therefore, we select the *absolute cost* $c(p)$ as our first evaluation metric.

Transmission planners have to decide whether to allocate data tokens in the planning horizon or not. Non-Allocation can either mean that the data token has been dropped or that it is delayed beyond the current planning horizon. The drop and long-term delay rate is an important indicator to check whether the analyzed strategy balances the attracting and repelling forces well or not. We define this dependent variable by the share of overall data tokens that have been allocated in the current planning horizon.

Transmission Planning relies on the network environment and data to transmit. This data is not available long before transmission. Significant input changes make recalculation inevitable. To be able to apply transmission plans, the planning process has to be responsive and fast in the calculation. Hence, we select transmission planner *execution time* as our second evaluation metric.

4.3.1.1 *Normalized Rating Score*

Absolute cost $c(p)$ depends strongly on the scenario. For example, in a scenario with bad network connectivity, application QoS requirements cannot be satisfied. This dependence may result in a high cost value even in an optimal transmission plan. Hence, the evaluation over multiple scenarios cannot be subsumed with absolute cost. To achieve independence from the scenario, we introduce a new relative metric: Normalized Rating Score (NRS). NRS describes a transmission plan's used share of the absolute optimization potential of the given scenario. A value of 0.8 means that a transmission plan uses 80% of the scenario's optimization potential. To define the optimization potential, we employ an upper and a lower bound.

NRS normalizes the transmission planner's performance to the absolute optimization potential of the given scenario. It enables comparison and statistics over multiple independent scenarios.

As lower NRS cost bound, we select the lowest possible cost value c_{Opt} of the scenario. To identify this cost, we implemented an optimization as an Integer Linear Program (ILP) that minimizes the cost function using the IBM CPLEX Branch&Bound solver. As upper cost bound for the relative metric, we select the average cost of a best-effort random transmission planner c_{Rnd}. We consider this as a reasonable upper cost bound since no transmission planner should perform worse than random. The exact algorithm to create random but valid transmission plans is given in Appendix A.2.

Equation 4.29 normalizes the absolute cost c_p of a transmission plan p to these bounds. Instead of showing absolute cost, Normalized Rating Score presents the share of the reached absolute optimization potential. Its value is 1 for optimal plans and may be negative for plans with a cost lower than that of average random plans. Hence, NRS enables direct comparison of transmission planner results over multiple scenarios. We use *Normalized Rating Score* (NRS) as our third evaluation metric.

$$\mathrm{NRS}(p) = \frac{c_{Rnd} - c_p}{c_{Rnd} - c_{Opt}} \tag{4.29}$$

4.3.1.2 *Relative Optimization Potential*

To gain deeper insights of the performance characteristics presented using NRS, it is instructive to analyze the therein defined absolute optimization potential as well. The absolute optimization potential is the cost value difference between the average cost value of a random transmission planner and an optimal transmission plan, as shown in the denominator of Equation 4.29. However, the values from this difference are hard to interpret as they still rely on the scenario.

Therefore, we introduce the Relative Optimization Potential (ROP), as presented in Equation 4.30, normalizing the optimization potential to the average random cost. Hence, for an average random cost equal to the optimum, the Relative Optimization Potential cost is 0, representing the lower bound. In contrast, the Relative Optimization Potential's supremum is 1 for the case that the optimal cost is 0. We especially use the dependent variable ROP to analyze the characteristics of the NRS results. ROP is not an evaluation metric applied to the planning strategies. Instead, it characterizes the scenario and is used as an indicator for correlating effects.

$$\mathrm{ROP} = \frac{c_{\mathrm{Rnd}} - c_{\mathrm{Opt}}}{c_{\mathrm{Rnd}}} \tag{4.30}$$

4.3.1.3 *Relative Detail Score*

Absolute cost and Normalized Rating Score provide information about the overall performance of a transmission plan. However, for a deeper analysis, it is additionally interesting to identify the strengths and weaknesses of transmission plans. For this reason, we introduce Relative Detail Score (RDS). It presents the relative cost difference for one cost category v compared to this of to the optimal plan. Hereby, we use the main cost sources from the defined rating function as categories: violation of deadline and start time, requirement violation of the minimum throughput, latency, jitter, allocated tokens and monetary cost. For each of these criteria v, the difference between the cost of the analyzed transmission plan p and the one from the optimal plan is divided by the absolute cost difference of the two plans. A value of 0 means that the cost share of transmission plan p for criterion v is equal to this of the optimal transmission plan. A value greater than 0 reveals a higher cost share. It means that the transmission plan p creates a higher cost for criterion v and reveals, that the planner should act more restrictive on it. In contrast, a value smaller than 0 is a sign for a too restrictive behavior of the transmission planner in criterion v. The designer of the transmission planner should consider making the model on criterion v less restrictive in order to create room for improvement in other criteria. The corresponding definition is given by equation 4.31. Hence, the Relative Detail Score provides the means to do detailed analysis on a transmission planner's strengths and weaknesses. We select *Relative Detail Score* as our fourth evaluation metric.

RDS normalizes cost shares of a transmission plan to those of an optimal transmission plan. It can be used to identify strengths and weaknesses of transmission planners.

$$\mathrm{RDS}_{v}(p) = \frac{c_{v,p} - c_{v,\mathrm{Opt}}}{c_{p} - c_{\mathrm{Opt}}} \tag{4.31}$$

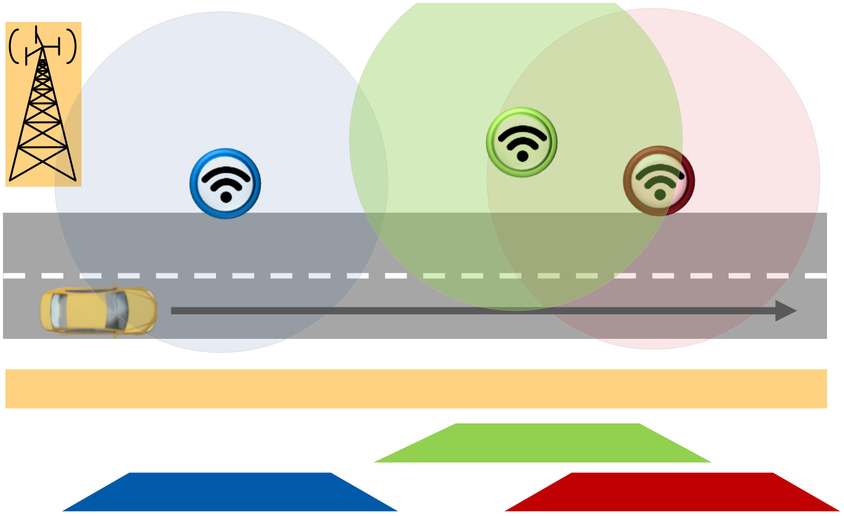


Figure 8: Example scenario with one mobile network and three WiFi networks

4.3.2 *Evaluation Setup and Independent Variables*

For evaluation, we generate and simulate randomized connected vehicle scenarios covering multiple networks, created subject to constraints as explained in this section. Generated scenarios contain characteristics of mobile networks and WiFi networks in ratio 1:3, employing randomly corresponding characteristics from 2G, 3G or 4G mobile networks, respectively, 802.11p, 802.11n or 802.11ac characteristics, as detailed in Section 2.2. An example with four networks is illustrated in Figure 8, showing a mobile network in yellow and three WiFi networks in green, blue and red along the street. From the vehicle movement, the network availability over time and its characteristics are derived. This availability over time is illustrated using the colored bars at the bottom. Each network has a certain availability and characteristics, i.e. network capacities as bucket sizes $B_{n,t}$ for each network in each time slot as well as latency, jitter and monetary cost properties.

The data traffic consists of four classes: interactive (5%), conversational (15%), bufferable (55%) and background (25%). These traffic classes represent default categories. We give a detailed definition of the four traffic classes with the corresponding requirements in Appendix A.3. Their share follows the mobile Internet consumption analysis and prognoses from Sandvine[187] and Cisco [17]. In contrast to smartphones, the connected vehicle is supposed to sense the road environment and provide the data for driver assistance systems and automated driving features [146], improving their service with aggregated external up-to-date information [35]. To cover this scenario-specific up- and download data transfer, we select a higher share for background traffic than for smartphone users. Instead of about 15% for background data, we use 25% in our simulation. We consider this as more realistic, even though the effects on the result are marginal. As background data transmission is usually delay-tolerant, strategies implementing time selection, i.e. Delayed Wifi Offloading, Opportunistic Network Selection and Joint Transmission Planning, profit from this change, highlighting the importance of transmission

Table 3: Independent simulation parameters

Time slots (T)		25	50	**100***	200	400
Networks (N)	1	2	4	**8***	16	32
Data flows (F)			4	**8***	16	
Data traffic load ($load$)			low	**medium***	high	
Monetary cost weight (ω_{mon})		zero	low	**medium***	high	

*default parameter values in the fractional factorial evaluation
The actual parameter values for zero, low, medium and high are detailed in the corresponding sections.

time selection in the scenario. We configure the client with one mobile network modem and one WiFi interface.

In the following, we identify the independent variables. The scenario may vary in its size, firstly in the planning time horizon and the planning granularity. The granularity is defined abstractly through the number of time slots T and the time slot duration $\Delta\tau$, as detailed in Section 4.1. Since T directly influences the problem complexity, we select it as an independent variable and vary it between 25 and 400 time slots. As a second independent variable, we identify the number of available networks N within the given scenario. Since all presented algorithms target network selection, the number of networks N should have an impact on all strategies. We expect a higher optimization potential in environments with high network diversity. Hence, we select N between 1 and 32 available networks in a given planning horizon. Thirdly, the number of data flows F influences granularity of planning and the variety of network matches. We vary F between 4 and 16 data flows. We select 4 as minimum to realize the data traffic share of the four defined categories in each scenario. We separate it from a fourth independent variable of data traffic load, which we investigate on its own. We expect that the strategies behave differently in sparse or overloaded data traffic scenarios. In connection with the network capacity, the pair of the number of data tokens $\widehat{p}_f$ and the token size d forms a similar duality as T and $\Delta\tau$: A high token count $\widehat{p}_f$ either can represent high data traffic or, if the token size d is small, a high planning granularity. Finally, we evaluate the impact of the willingness to pay, i.e. the monetary cost weight ω_{mon}. We vary it from zero to high. It balances the two main minimization objectives of the rating function on flow QoS requirement violation and monetary cost. With varying this parameter, we investigate the impact of the two objectives on the presented transmission planning strategies.

Table 3 summarizes the five identified parameters with its values. For evaluation, we use a fractional factorial evaluation design [25]. Hence, we vary only one single parameter, keeping the other parameters as controlled variables constant at a dedicated default value. In Table 3, the default values are highlighted and marked accordingly. In the following, we analyze the absolute cost, the execution duration, NRS, ROP, RDS and the data drop rate for each parameter variation and show the $Q_{25\%}$, $Q_{50\%}$ (median) and $Q_{75\%}$ quantiles for 50 randomized scenarios per run.

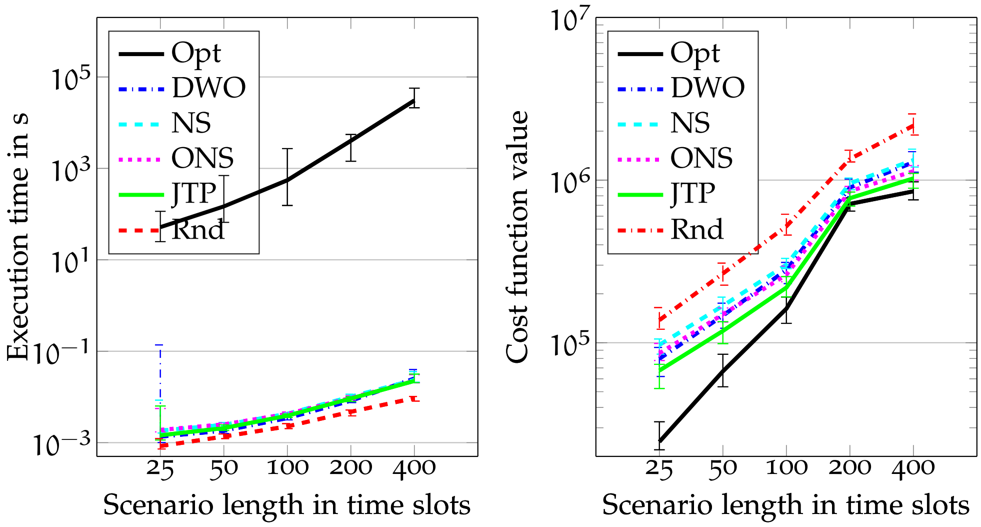


Figure 9: Execution time (left) and absolute cost (right) over scenario length in time slots

4.3.3 *Impact of Time Selection: Variation of the Planning Horizon Length*

A main goal of this thesis is understanding the effects of transmission time selection in joint time-network selection. To evaluate the transmission time selection impact of the planners on perceived transmission quality, we evaluate the presented algorithms in scenarios with different planning horizon length. A scenario with a longer planning horizon provides a higher potential for moving delay-tolerant data transmission in the time dimension. This effect is also shown by Lu [137] and Lee [132]. Note that a larger number of time slots can either correspond to a larger temporal scenario planning horizon or to a higher granularity of time slots. We vary the planned scenario planning horizon length from 25 to 400 time slots and use the above-mentioned default values for all other independent variables. In the following, we present and discuss the effects on the dependent variables using the metrics presented in 4.3.1.

EXECUTION TIME. The execution time of the presented algorithms rises with the planning horizon length, as shown in Figure 9 left, showing slightly exponential characteristics. There is no substantial difference between the execution times of the heuristic approaches. All values for calculation a scenario length of 400 time slots are below 0.05s , indicating the real-world applicability of the algorithms. In contrast, the optimization approach requires about four to five orders of magnitude more time to calculate the result. This long processing time disqualifies optimization for real-world use and confirms the need for efficient heuristics. However, we still use the optimization for evaluation purpose as an upper quality limit.

ABSOLUTE PERFORMANCE. The cost function plot in Figure 9 right shows about linear rising cost function values for scenarios with longer duration. This effect is expected since the average data traffic from the scenario generation process is

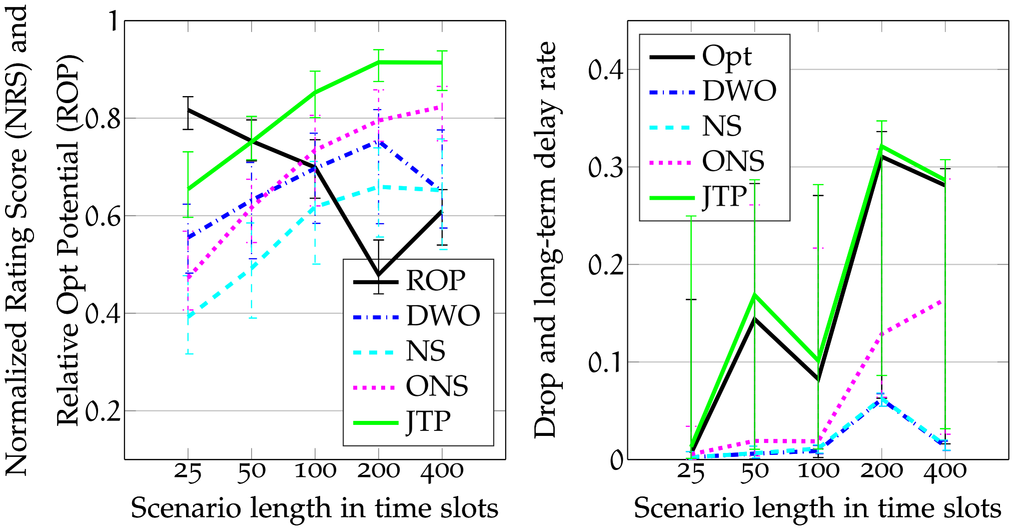


Figure 10: Normalized Rating Score (NRS) (left) and Drop and long-term delay rate (right) over scenario length in time slots

Table 4: Median NRS gains of JTP over number of time slots

Time slots	25	50	100	200	400	mean
Gain over NS in %	26.22	25.90	23.45	25.53	26.18	25.45
Gain over ONS in %	18.19	13.46	11.83	11.95	**9.06**	12.90
Gain over DWO in %	9.92	11.96	15.57	16.16	**26.51**	16.02

about constant for each time slot. Longer scenario duration, i.e. a higher number of time slots, conclusively leads to more data traffic and therefore to higher cost with same quality. However, the effects are not observable from absolute values, as the confidence intervals are strongly overlapping because results strongly depend on the dedicated scenario of each repetition. To receive a better understanding of the performance of the approaches, we use Normalized Rating Score (NRS) in the following, which cancels out scenario dependence.

RELATIVE PERFORMANCE AND OPTIMIZATION POTENTIAL. The Normalized Rating Score results provide insights on how much of the scenario's optimization potential the approaches are able to use. They are shown in Figure 10 and detailed in Table 4. The results of the approaches integrating a network selection show a strict order in their NRS results, rising according to their time selection strength. There exists a positive correlation between the degree of time selection of the approaches and their quality. However, time selection alone, represented by DWO, is dominated on the long run.

Indeed, the Joint Transmission Planning (JTP) reaches the best results, exploiting in median up to 91.45% NRS, i.e. the scenarios' absolute optimization potential. The median results show an average gain of 25.45% over Network Selection (NS)

Table 5: T-test results for $H_0 : \overline{\text{JTP}} = \overline{\text{ONS}}$ and $\overline{\text{JTP}} = \overline{\text{DWO}}$ over number of time slots

Time slots	25	50	100	200	400
$p_{\overline{\text{JTP}}=\overline{\text{ONS}}}$	$1.3 \cdot 10^{-10}$	$1.2 \cdot 10^{-10}$	$4.8 \cdot 10^{-10}$	$1.4 \cdot 10^{-10}$	$1.2 \cdot 10^{-5}$
$p_{\overline{\text{JTP}}=\overline{\text{DWO}}}$	$2.4 \cdot 10^{-9}$	$1.3 \cdot 10^{-10}$	$3.9 \cdot 10^{-13}$	$1.4 \cdot 10^{-10}$	$1.1 \cdot 10^{-13}$

and 12.90% over the opportunistic approach (ONS) and 16.02% over Delayed WiFi Offloading (DWO). To confirm the statistical significance of these results, we exemplary convey a T-test with the null hypothesis that the results of Joint Transmission Planning and Opportunistic Network Selection, respectively Delayed WiFi Offloading originate from distributions with the same mean value, $H_0 : \overline{\text{JTP}} = \overline{\text{ONS}}$, respectively $H_0 : \overline{\text{JTP}} = \overline{\text{ONS}}$. Table 5 shows the p-values. As they are all far below 0.01, the null hypotheses can be rejected, indicating that Joint Transmission Planning significantly outperforms ONS and DWO, representing state-of-the-art derived approaches with the upper-bound performance of their respective strategies.

JTP significantly outperforms the other approaches by 9-26% for different scenario lengths.

However, we observe a common trend in NRS for all transmission planners: Their NRS shows a positive correlation with the simulated scenario length. We suppose that the effect depends more on the scenarios' optimization potential than on the actual performance of the transmission planners because it influences the three heuristic planners implementing network selection in the same way. Indeed, we observe in Figure 9 right that the difference of absolute cost between the optimal and the random approach shrinks with rising planning horizon. This difference defines the scenario optimization potential. We present it in a normalized form as Relative Optimization Potential (ROP), as specified in Section 4.3.1, in Figure 10 left as a black line. It shows a decreasing trend with rising planning horizon length. We expect this effect to originate from the following dependency: Increasing the number of time slots but keeping the number of networks constant, decreases the density of available networks. Moreover, keeping also the number of data flows constant while increasing the planning horizon, reduces the probability that the time limits of a data flow overlap with many networks. Thus, the opportunities for moving a data flow in time dimension to use another network diminish. Accordingly, the instances of the transmission planning problem with longer time horizon get easier to solve. The decreasing optimization potential causes the effect of the increasing relative performance of all three heuristic approaches with network selection over time horizon length. The behavior of DWO supports this statement, showing a strong negative correlation to the scenario optimization potential. A decreasing number of advantageous opportunities to move data in time increase the importance of network selection and render DWO without sophisticated network selection less effective. We analyze the effect of the scenario optimization potential on the transmission planner performances further during this evaluation.

STRENGTH AND WEAKNESS ANALYSIS. To identify strengths and weaknesses of the approaches, we have a deeper look into the resulting cost distribution of the four strategies, comparing it to the cost distribution of an optimal transmission

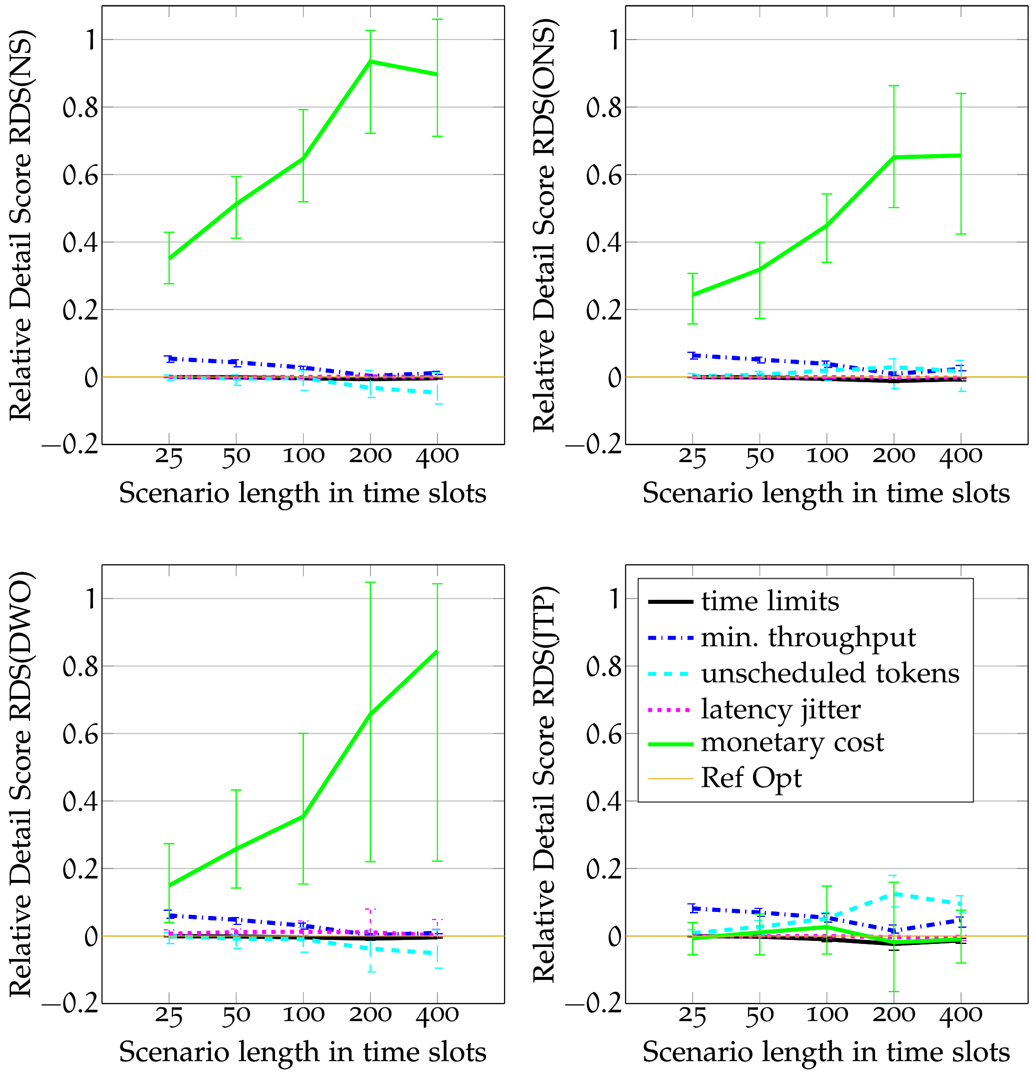


Figure 11: Relative Detail Scores (RDS) of the approaches NS, ONS, DWO and JTP over scenario length in time slots

plan. Therefore, we employ the Relative Detail Score (RDS), as defined in Section 4.3.1.3. RDS is positive if the cost share of the approach's transmission plan for a criterion is higher than thus of the optimal transmission plan. Hence, a positive value reveals that the potential for this criterion cannot be completely exploited, either because of a non-optimal parameter choice or because of an insufficient model. The same holds for negative RDS values. A negative value shows that the approach acts too restrictive to this criterion. A less restrictive handling could release additional optimization potential for other criteria, which may be reached either using a more accurate model or due to parameter optimization. Hence, we use the RDS profile to identify the strengths and weaknesses of transmission planners and unveil imbalances between two criteria, resulting from non-optimal trade-off parameter selection. Releasing the restrictiveness from one criterion through parameter changes can give more freedom to another one and may improve the overall performance. Hence, in the case that all values are above zero, it is improbable that trade-off parameter changes can significantly improve the performance of a strategy. In contrast, such imbalances indicate general weaknesses of the strategy.

Best-effort transmission of data impairs the performance of NS and DWO.

As presented in Figure 11, all strategies show closeness to the optimum for the QoS requirement satisfaction of the data flows in time limits, throughput, latency, jitter and unscheduled tokens. However, monetary cost characteristics show a significant difference: The three state-of-the art strategies cause a much higher monetary cost, leading to a huge imbalance in the characteristics, rising with the scenario length in time slots. In addition, we observe a rising variance of the RDS monetary cost values with rising scenario length in time slots, especially for Delayed WiFi Offloading, showing that their performance relies strongly on the given scenario. The imbalances for the state-of-the-art approaches reveal that there exist substantial weaknesses in the strategies. It shows that the approaches cannot exploit the full optimization potential, selecting in general too expensive networks. For larger scenarios, values slightly below zero can be observed for unscheduled tokens for NS and DWO, which transmit more tokens because of employing a best-effort data allocation. Introducing the opportunistic delaying of ONS, we can observe a significantly lower RDS value for monetary cost, while unscheduled tokens do not sink below zero. This phenomenon explains the benefit of ONS over NS, emerging from its decision opportunity not to transmit all data within the current time horizon. However, there are no other RDS values significantly below zero. Hence, the RDS profile reveals that presence of defects in transmission planning strategies.

In contrast, Joint Transmission Planning shows low and overall balanced RDS values. Notable impacts on the minimum throughput and unscheduled token criteria indicate that the strategy may still be improved in its allocation decision. In addition, we observe the marginal trend of a falling RDS for time limits in long-term scenarios. It results from the fact that our heuristic approaches do not consider data token allocation beyond the desired time frame between start time and deadline. As the off-trading criteria unscheduled tokens already shows low RDS values, we expect an additional heuristic for time limit violation handling to result in negligible effects.

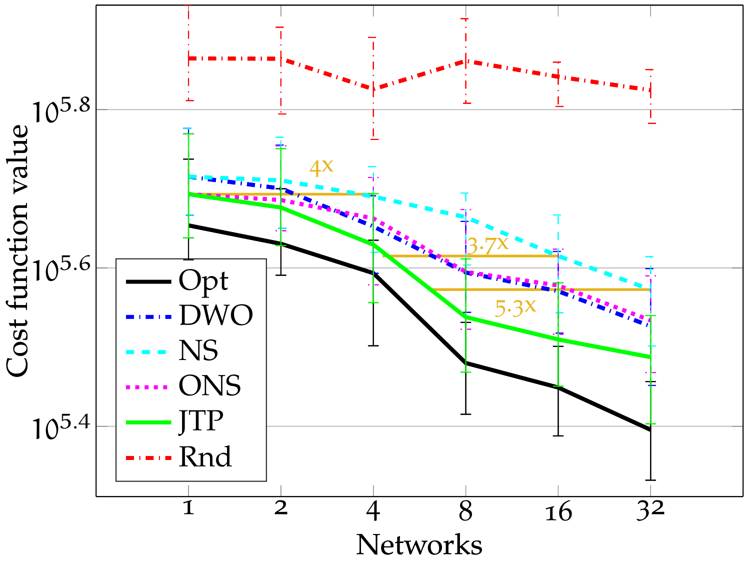


Figure 12: Absolute cost value over number of networks

CONCLUSIONS. The evaluation of the four strategies reveals significant benefits from integrating transmission time selection and network selection. In fact, Joint Transmission Planning (JTP) outperforms the approaches derived from state-of-the-art by 25.45%, respectively 12.90% and 16.02% NRS. However, we observe a similarly increasing performance of all transmission planners with time selection over an increasing planning time horizon. As explained above, several effects cause a decreasing optimization potential. We assume the two negatively correlated effects, increasing performance of the strategies and a sinking optimization potential, to be linked. Accordingly, the transmission planners achieve relatively better performance in scenarios with a lower optimization potential. We investigate this assumption further the following evaluation.

4.3.4 *Impact of the Number of Networks*

Varying the number of networks provokes network selection to take effect, with no network selection impact in the case of a single network, and significant network selection impact in the case of many networks. In fact, using a single network scales down the solution space of our approaches to that of pure, single-homed transmission time selection, reflecting important approaches of Bui et al. [26] and Lu et al. [137]. We vary the number of networks between 1 and 32 and analyze the scenario optimization potential and corresponding effects. Moreover, we compare the joint time-network selection strategy to pure single-network transmission time selection and pure network selection.

JOINT TIME-NETWORK SELECTION BENEFIT The absolute cost value of all approaches decreases significantly with an increasing number of networks in the scenario, as shown in Figure 12. This effect confirms that a higher number of net-

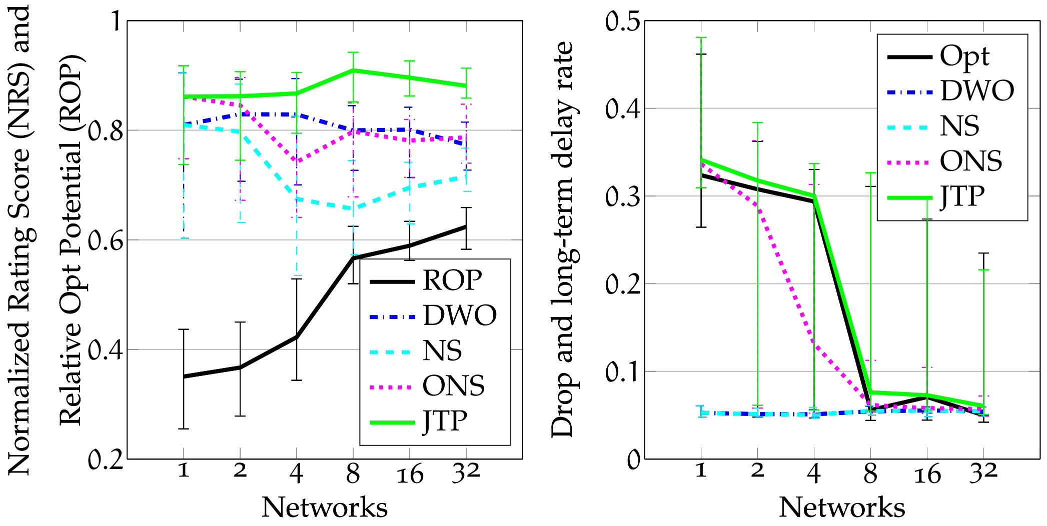


Figure 13: and NRS (left) and drop and long-term delay rate (right) over the number of networks

works provides a higher potential for network selection as well as for transmission time selection.

Joint time-network selection improves the perceived transmission performance. The gain is similar to using pure network selection in an environment with 4 times more networks.

Comparing the performance of JTP with a single network to the same performance level of NS, we discover a factor 4 of additional networks, marked with a golden horizontal line in Figure 12. Hence, the performance of JTP in an environment with a single network is equal to that of using NS in an environment with 4 times more networks. Analyzing this factor for 16 and 32 networks of NS, we observe that the factor 4 stays similar, reaching 3.7 with 16 networks, respectively 5.3 for 32 networks, both marked with golden horizontal lines in Figure 12.

We assume that joint time-network selection reaches a similar performance as a network selection in an environment with about 4 times more networks. However, we cannot prove the general validity of the factor 4, which might be different significantly different environments. Nevertheless, the analysis reveals a significant trend that highlights clear benefits of joint time-network selection.

RELATIVE PERFORMANCE ANALYSIS The transmission planners with a sophisticated network selection show the same strict order for variation of the number of networks as for time horizon length variation. This confirms the benefits of Joint Transmission Planning over the state-of-the-art derived approaches.

JTP outperforms the other approaches significantly in scenarios with at least 4-8 networks within the planning time horizon.

JTP achieves a median performance of up to 90.93% NRS, outperforming in average the approaches NS by 15.47%, ONS by 7.71% and DWO by 7.26%. However, the benefits over ONS and DWO start getting significant ($p < 0.01$) at 4, respectively 8, networks in the scenario horizon, as visible from Table 6. Furthermore, the Normalized Rating Scores of NS and ONS show a U-shaped characteristic, visible in Figure 13 left, with a drop at 4 networks of 10.34%, respectively 12.27%, and recovering with a rising number of networks. In contrast, JTP show a stable performance, demonstrating its superiority and robustness against variation of the network environment.

Table 6: T-test results over number of networks

Networks	1	2	4	8	16	32
$p_{\overline{JTP}=\overline{ONS}}$	**0.81**	**0.47**	$3.7 \cdot 10^{-6}$	$1.8 \cdot 10^{-7}$	$2.8 \cdot 10^{-10}$	$1.9 \cdot 10^{-7}$
$p_{\overline{JTP}=\overline{DWO}}$	**0.74**	**0.43**	**0.14**	$3.2 \cdot 10^{-7}$	$2.6 \cdot 10^{-8}$	$3.5 \cdot 10^{-9}$

The Relative Detail Score and execution time results show similar characteristics as for the analysis of the variation of time slots. We detail these results in the Appendix A.5. In the following, we investigate reasons for the insignificance of JTP's gain in environments with few networks and the discovered U-shape in ONS and especially NS.

OPTIMIZATION POTENTIAL AND RESOURCE SATURATION. For the random approach, representing the upper bound for optimization potential definition, we observe an about constant absolute cost value for 1 to 8 networks, as illustrated in Figure 12. For 16 and 32 networks, the cost value of the random approach starts to sink. For all other approaches, the absolute cost function value sinks faster than linear, whereby the value of the optimum transmission plans decreases fastest. From the falling characteristic of the random approach in absolute cost value for many networks, there originates an interesting characteristic in NRS: The NRS over the number of networks shows a U-shape for NS and ONS with a minimum at 4 networks, cf. Figure 13 left. This U-shape in NRS results from two superposed effects: First, a rising number of networks increases the opportunity to be able to move data to other networks that are potentially available at another point in time. Hence, increasing the number of networks increases the optimization potential, especially for transmission time selection. This effect is also reflected by slightly rising NRS values for JTP and DWO from 1 to 4 networks, while the performance of NS and NRS drops in the same region. However, beyond that, there starts a saturation of good networks, which causes the second effect. With a high number of networks, the probability of excellent networks being available increases. Thus, there is a good chance that for most of the time excellent networks are available, which satisfy most application QoS requirements and are not too expensive. Due to this saturation of excellent networks, the transmission time selection impact loses importance and network selection strategies perform better, reflected by rising NRS values for NS and ONS. Accordingly, the NRS values for DWO start sinking at the same point because DWO lacks an appropriate network selection and cannot select the best-suited of the available ones. To show this saturation of network resources, we analyze the drop and long-term delay rate of the different strategies over the number of networks, as illustrated in Figure 13 right. It illustrates what shares of data tokens are either dropped or moved for transmission beyond the planning horizon. It confirms the saturation statement of excellent network resources. For the optimal schedule, the median drop rate decreases from about 30% to less than 10% when reaching the saturation limit of 8 networks. Thus, the two superposed effects result in a falling performance in NRS of state-of-the-art network selection approaches with no or limited time selection strategies, while

A single-network scenario provides no choice of networks. With lots of networks, there is a saturation of excellent resources. In both extreme cases, network selection performs well compared to JTP.

at the same time the presence of an appropriate network selection mechanism in transmission strategies gains importance. For scenarios with few networks, the ROP in Figure 13 indicates a low optimization potential, reflecting that selection is in many cases not possible because there are no options. All heuristic approaches perform well, as there is no significant challenge in data allocation. Therefore, JTP develops its significant performance gain only in network environments with several alternative networks in the planning time horizon. Nevertheless, constantly best results of JTP and a strong robustness against different network environments highlight the importance of integrating both aspects, transmission time selection and network selection.

CONCLUSIONS. Varying the number of networks provides the opportunity to compare joint time-network selection to pure single-homed time selection and pure network selection. It reveals the following important finding, which holds at least for the given scenario: *Joint time-network selection provides a performance gain, which is similar to using pure network selection in an environment with 4 times more networks.*

In addition, the importance of transmission time selection is highest in environments with few alternative networks, reflecting the case that, firstly, there exist options for networks to select from and, secondly, there is no saturation of excellent network resources yet, being available at any time. In contrast, network selection gains importance as soon as there are many networks from which to choose. Hence, the NRS performance of Network Selection (NS) and Opportunistic Network Selection (ONS) sinks to a minimum in environments with few networks, while the performance of Delayed WiFi Offloading (DWO) decreases towards saturation of network resources. Finally, Joint Transmission Planning (JTP), integrating both aspects in its time-network selection strategy, mitigates these impacts. It outperforms the state-of-the-art approaches by 15.47% (NS), 7.71% (ONS) and 7.26% (DWO), proving its superiority due to robustness against different network environments with a constantly high performance of in average 87.96% NRS.

4.3.5 *Impact of the Data Traffic Load*

In this section, we analyze how the amount of traffic of a mobile node influences the performance of the different strategies. While keeping the network resources and the number of flows constant, we vary the amount of data traffic in tokens to be sent. Note that tokens represent abstract data units for planning with a token size d. A high number of tokens can either correspond to a high amount of data or a low token size d. We vary the number of tokens in three stages, according to Table 7, keeping the token sizes d constant. We define the amount as the number of tokens per time slot and distribute the number of tokens among the 8 data flows in our scenarios, according to Appendix A.3. Varying the data amount reveals interesting effects that we present in the following.

PERFORMANCE AND RESOURCE SATURATION The transmission rating's absolute cost function value reveals a rising characteristic with the number of tokens, illustrated in Figure 14 left. As tokens create cost in the rating model in any case,

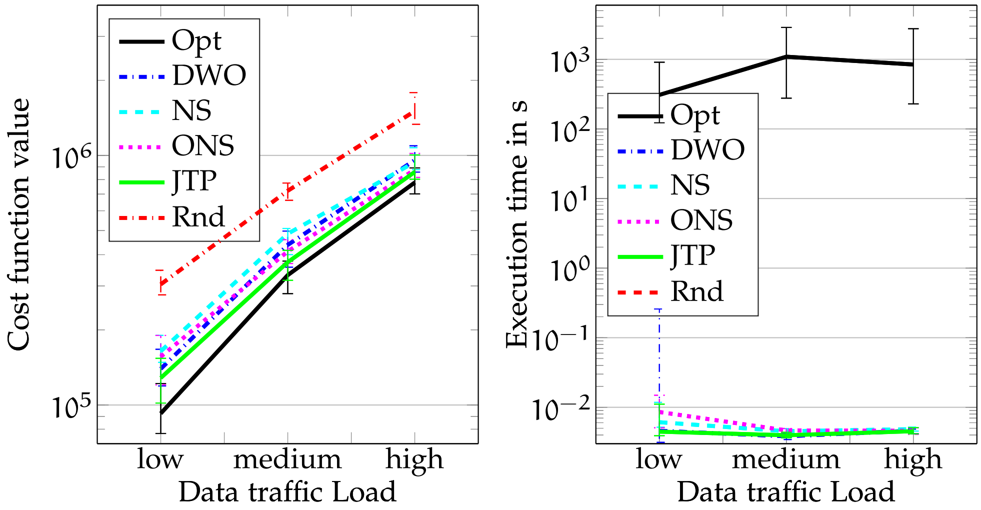


Figure 14: Absolute cost value (left) and execution time (right) over data traffic load

contributing to repelling forces if allocated and contributing to attracting forces if not, a rising number of data tokens comes with a rising cost function value. In addition, a higher data traffic load leads to an insignificant increase of the execution time for all approaches, as illustrated in Figure 14 right.

However, an analysis of the Normalized Rating Score (NRS) and the drop rate from Figure 15 gives a clear insight into the transmission planner characteristics. The NRS of the network selection based state-of-the-art derived approaches NS and ONS shows a rising characteristic and about constant, tendentiously sinking, performance for DWO. DWO suffers significantly (p<0.01) from the lack of a sufficient network selection, while in scenarios with low data traffic load, it is able to push most transmissions to WiFi networks, reaching a considerable NRS value of 79.01%. Starting from a low NRS level for NS of 65.46%, respectively 75.34% for ONS, the performance rises to 77.58%, respectively 85.75%, while the performance stays between 76.21-79.01% NRS for DWO. Again, the rising characteristic of NS and ONS correlates negatively with the scenarios' Relative Optimization Potential (ROP). We suppose that this decreasing optimization potential originates from two correlating effects. Firstly, networks are used more intensively in general. Therefore, it is less critical to decide for the best matching or cheapest networks over time because nearly all possible resources are used. Secondly, the data drop rate rises significantly from less than 10% to more than 30%, as visualized in Figure 15 right, reflecting that there are not sufficient appropriate network resources for allocation of all data tokens. Therefore, the share of data which is dropped because of a lack of resources rises in comparison to the share for which a dedicated allocation

JTP outperforms the other approaches. However, at high data traffic loads, time selection loses importance because unused transmission opportunities to select from are rare and cannot be skipped.

Table 7: Traffic load parameters in tokens per time slot

Traffic load	low	medium	high
Tokens per time slot	30-60	120-150	270-300

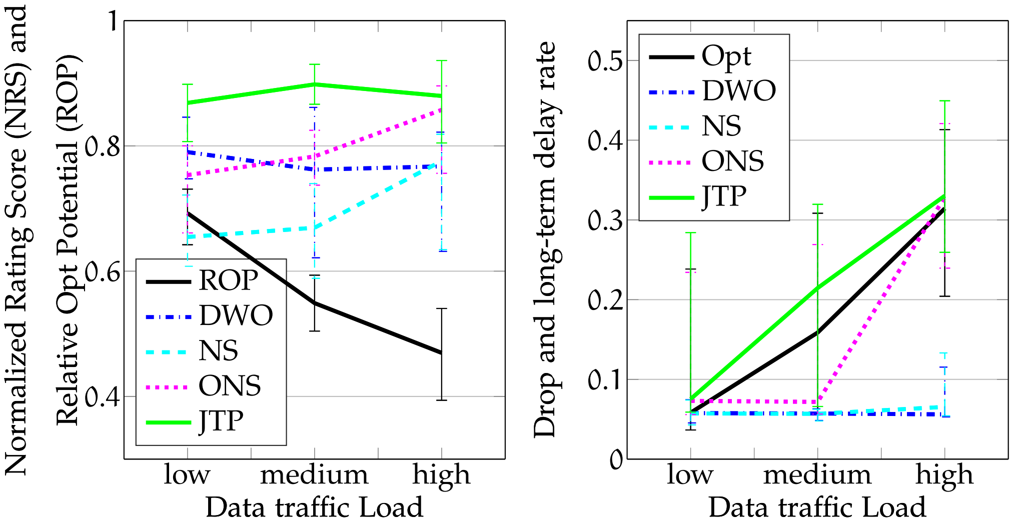


Figure 15: NRS (left) and drop rate (right) over data traffic load

Table 8: T-test results over data traffic load

Data traffic load	low	medium	high
$p_{\overline{\text{JTP}}=\overline{\text{ONS}}}$	$1.0 \cdot 10^{-6}$	$6.7 \cdot 10^{-11}$	**0.096**
$p_{\overline{\text{JTP}}=\overline{\text{DWO}}}$	0.006	$1.2 \cdot 10^{-10}$	$4.9 \cdot 10^{-7}$

decision from the strategy is required. This effect creates a higher match between the transmission plans of the heuristic approaches and those of the optimal one. Finally, for high data traffic load, ONS reaches a level with an insignificant (p=0.096) difference to JTP, as shown in Figure 8. In contrast, Joint Transmission Planning can cope well with any amount of data traffic, shows 86.87-89.82% median NRS and outperforms the approaches derived from state-of-the-art by 18.23% (NS), 8.41% (ONS) and 10.90% (DWO).

Furthermore, the Relative Detail Score analysis in Figure 16 confirms identified effects and characteristics of the strategies. For ONS there are no RDS values significantly below zero, while NS and ONS, transmitting in best-effort fashion, show negative RDS values for unscheduled tokens, accompanied by exploding monetary cost. This imbalance confirms the strategical defect of those approaches. In contrast, Relative Detail Score reveals that Joint Transmission Planning tends to delay too much data beyond the planning horizon, indicated by a significantly increased RDS value for unscheduled tokens and a negative monetary cost RDS value for low data loads. We try to cure this imbalance through parameter optimization of the schedule decision, as shown in Appendix A.4. Even though the characteristics of the RDS results change as desired for this optimization, the optimization leads only to insignificant effects of the final result. Hence, the main difference to optimal schedules seems to result from strategical defects of the approach.

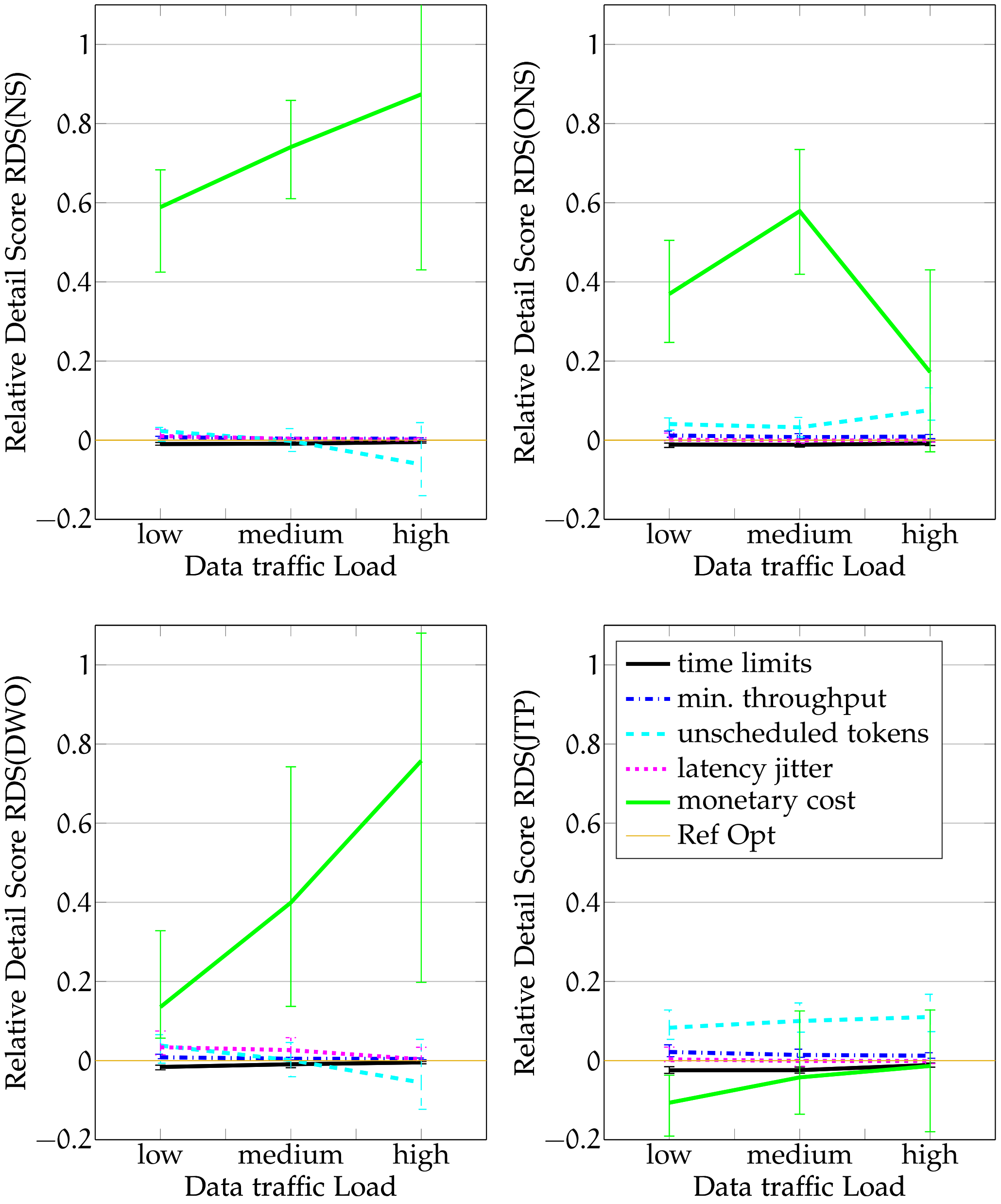


Figure 16: Relative Detail Scores of NS (top left), ONS (top right), DWO (bottom left) and JTP (bottom right) over data traffic load

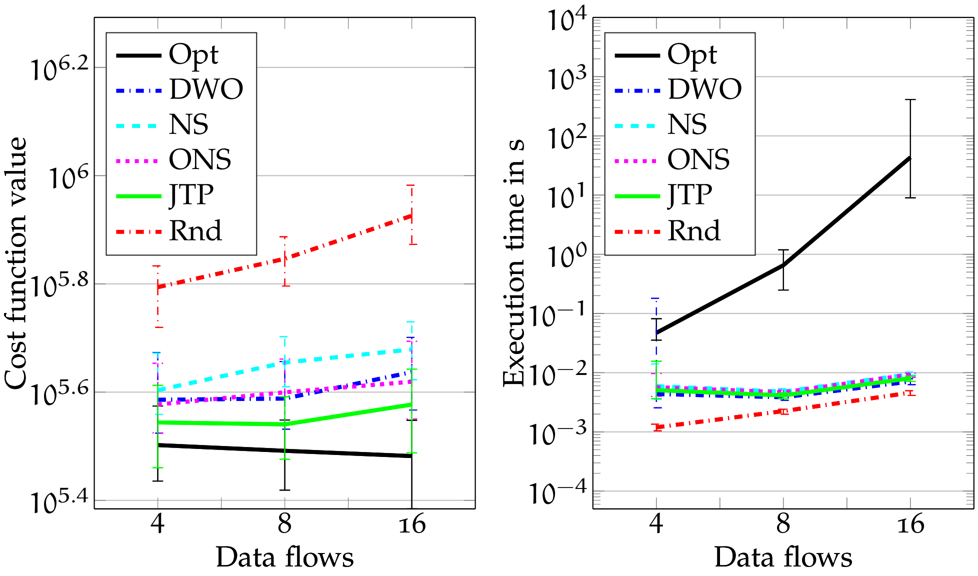


Figure 17: Execution time (right) and absolute cost (left) over number of data flows

CONCLUSIONS. Increasing the data traffic of a mobile node leads to a saturation of the available network capacity. Accordingly, the client uses most of the available resources over the whole trip duration, reducing the Relative Optimization Potential (ROP) of the scenarios. With rising data traffic load, transmission time selection loses importance because later-available resources will be fully used and no opportunities to transmit should be skipped. Accordingly, the NRS performance of the two state-of-the-art derived network selection approaches NS and ONS rises fast with increasing data traffic, while Delayed WiFi Offloading (DWO) shows about constant, tendentiously sinking performance. Nevertheless, Joint Transmission Planning (JTP) outperforms the state-of-the-art derived approaches in most cases significantly by 18.23% (NS), 8.41% (ONS) and 10.90% (DWO) NRS. It constantly achieves high NRS results between 86.87% to 89.82% median NRS, proving its robustness against variation of the amount of data traffic.

4.3.6 *Impact of the Number of Flows*

Incrementing the number of data flows increases the diversity of the data traffic, which increases the opportunities for data flow prioritization and balancing. We vary the number of data flows between 4 and 16, keeping all other parameters constant. This also includes the amount of data traffic. The fixed amount data traffic to be planned is distributed over the number of flows. Hence, with increasing number of data flows, their amount of data sinks. We select the lower value of 4 data flows corresponding to the number of our traffic classes and using less than 4 data flows does not allow to model a realistic data traffic mix. Details about our traffic classes are given in the Appendix A.3.

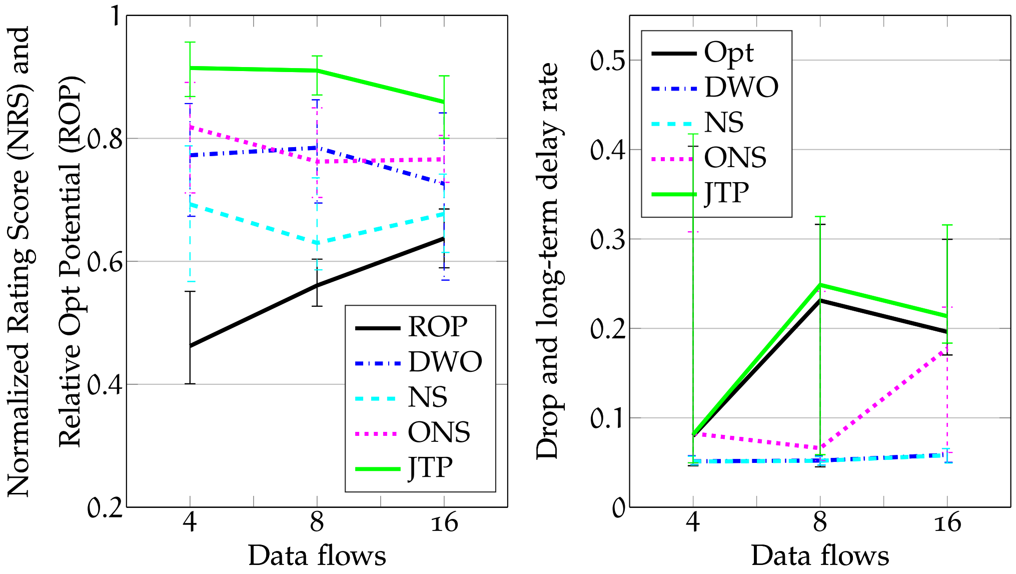


Figure 18: Normalized Rating Score (left) and Drop rate (right) over number of data flows

Table 9: T-test results over data flow

Data flows	low	medium	high
$p_{\overline{JTP}=\overline{ONS}}$	$8.6 \cdot 10^{-8}$	$1.5 \cdot 10^{-12}$	$3.0 \cdot 10^{-5}$
$p_{\overline{JTP}=\overline{DWO}}$	$9.7 \cdot 10^{-11}$	$5.2 \cdot 10^{-10}$	$3.9 \cdot 10^{-6}$

performance and optimization potential. In contrast to the other presented parameter analyses, the number of data flows barely changes the absolute cost function value, while the changes even trend into different directions, as visualized in Figure 17 left. While optimal plans reach a tendentiously lower absolute cost function value for more data flows, i.e. a higher requirement diversity in the data, it rises for all other approaches. The execution time of the heuristic approaches is nearly unaffected from the rising number of data flows (and constant amount of data), which is beneficial for scaling. In contrast, the execution time of the optimal approach rises severely by a factor of 66 for doubling the number of data flows from 8 to 16.

The Normalized Rating Score results in Figure 18 shows stable results for all approaches, with tendentiously lower NRS results for 16 data flows than for 4. This sinking relative performance confirms the earlier identified negative correlation with the Relative Optimization Potential (ROP). We suppose that a high number of data flows leads to an increasing diversity of data traffic, requiring more accurate prioritization and balancing of data flows. Especially the balancing in network selection is an interesting aspect. For the heuristic approaches, the network selection in a time slot depends on the preference of the data flows with the highest priority, i.e. the highest restrictiveness. However, in some cases, a common network preference of multiple lower prioritized data flows might dominate the data flow with

The number of data flows has a low impact on the performance of the approaches. JTP outperforms the other approaches significantly.

the highest priority. In the heuristic, we neglected these complex analysis dependencies with the expectations, that those conflicts can usually be solved in similar quality with time selection. Nevertheless, this effect can make a difference in the scenario with many competing data flows. Furthermore, the Relative Detail Score values show no exceptional characteristics and are, for completeness, presented in the Appendix A.5.2. Finally, JTP outperforms the other approaches significantly, as indicated by the t-test results in Table 9, in average by 22.81% (NS), 11.25% (ONS) and 13.35% (DWO), reaching an average NRS performance of 89.46%.

CONCLUSION. The number of data flows has a slight but insignificant negative impact on the performance and execution time of the heuristic approaches. For all runs, JTP significantly outperforms the state-of-the-art derived approaches by 11.25% (ONS) and 13.35% (DWO), reaching a robust average NRS performance of 89.46%.

4.3.7 *Impact of the Monetary Cost Weight*

Our rating function covers two main components for transmission planning. They are, firstly, application QoS violation and, secondly, monetary cost, whose minimization reflects contradicting objectives. The user's willingness-to-pay, defined as monetary cost weight ω_{mon} in the user preference model in Section 4.1.1, balances the the two components. Awad et al. investigated the [13] potential of dynamic objective weighting between transmission performance, monetary cost and energy consumption for their time selection approach. They demonstrate a significant benefit for adapting these weights according to the user's context, demonstrating the importance of analyzing the performance of the transmission planners with different objective balancing.

We investigate the effect of balancing these objectives on the transmission planner performance. Therefore, we vary the impact of the cost objective from zero to high. The corresponding monetary cost value weights ω_{mon} are shown in Table 10. The results are presented in the following.

PERFORMANCE RESULTS The monetary cost does not significantly influence the execution time of the heuristic strategies. However, the optimal transmission planner shows a lower execution duration whenever one of the objectives dominates, as illustrated in Figure 20.

The absolute cost values, presented in Figure 19 left, rise about linear over increasing the monetary cost weight for all transmission planners, resulting from the monetary cost which is added to the rating function without decreasing another one. When neglecting the monetary cost objective, Network Selection (NS) and Opportunistic Network Selection (ONS) reach similar performance as Joint Transmission Planning (JTP). This changes towards dominance of the monetary cost reduction objective. Indeed, the increasing monetary cost weight significantly impairs the NRS of the Network Selection approach, sinking from 87.83% down to 56.52% NRS, as illustrated in Figure 19 right. Accordingly, most application QoS requirements can be satisfied in the scenario with an instant transmission, if mone-

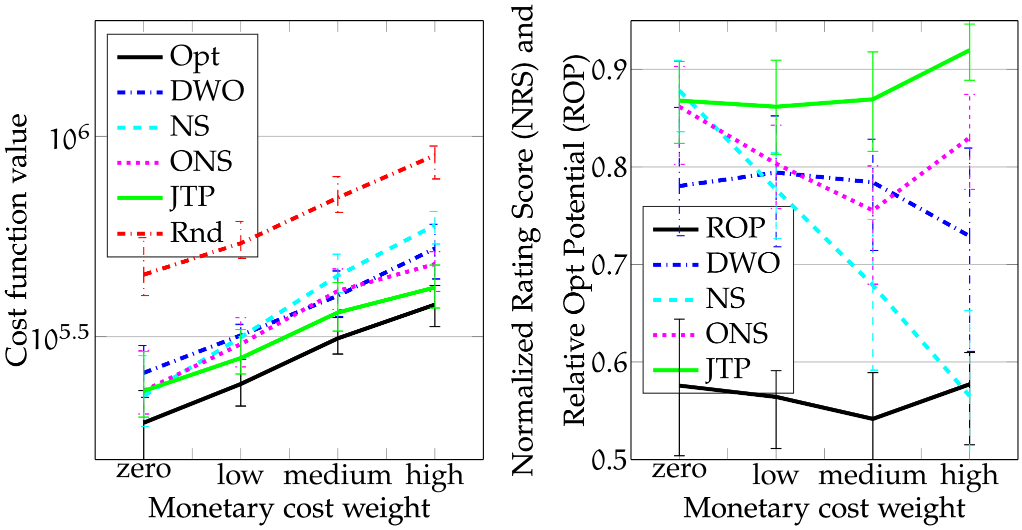


Figure 19: Absolute cost (left) and Normalized Rating Score (right) over monetary cost weight

Table 10: Monetary cost weight for objective balancing

Monetary cost weight	zero	low	medium	high
Weight ω_{mon}	0	5	15	25

tary cost does not matter. For rating, this does not matter as we consider quality as a relative metric, using the Normalized Rating Score (NRS). The latter characteristic gets obvious from the data drop rate. ONS and JTP, outperforming NS as soon as monetary cost is considered, cause a significant drop and long-term delay rate of up to 30%, meaning that transmission of delay-tolerant data is delayed beyond the planning horizon, while it is instantly transmitted in best-effort fashion by NS. In contrast, Delayed WiFi Offloading (DWO) shows about constant NRS performance till a medium monetary cost weight, distributing data more effectively over available networks in the planning time horizon than NS. Due to its lack of an appropriate network selection and best-effort transmission in the horizon, it reaches, in general, a lower and finally sinking performance because the monetary cost is only covered by the correlation of WiFi-preference and a statistically lower monetary cost for WiFi in the scenario generation, ignoring the actual monetary cost properties of the networks. For completeness, the RDS analysis is presented in the Appendix A.5.3, without showing unusal characteristics. Finally, JTP significantly outperforms the state-of-the-art derived strategies as soon as the monetary cost is different from zero, reaching average NRS performance gains of 21.00% (NS), 8.73% (ONS) and 11.44% (DWO). It constantly shows high NRS performance for all runs of 86.17% to 91.97%, proving its robustness against monetary cost weight variation.

JTP outperforms the other approaches significantly as soon as the monetary cost is considered. This shows that most requirements of transmissions can be addressed with instant, prioritized transmission when employing expensive networks.

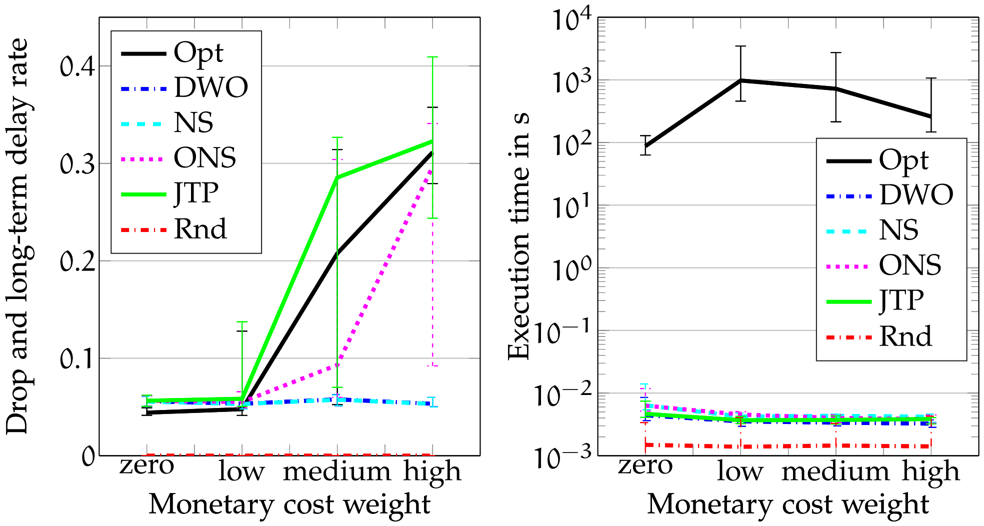


Figure 20: Drop rate (left) and execution time (right) over monetary cost weight

Table 11: T-test results over monetary cost weight

Monetary cost weight	zero	low	medium	high
$p_{\overline{\text{JTP}}=\overline{\text{NS}}}$	**0.132**	$8.6 \cdot 10^{-6}$	$1.0 \cdot 10^{-15}$	$6.7 \cdot 10^{-16}$
$p_{\overline{\text{JTP}}=\overline{\text{ONS}}}$	**0.689**	$1.2 \cdot 10^{-3}$	$1.3 \cdot 10^{-8}$	$2.0 \cdot 10^{-9}$
$p_{\overline{\text{JTP}}=\overline{\text{DWO}}}$	$2.6 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	$2.2 \cdot 10^{-5}$	$7.0 \cdot 10^{-16}$

CONCLUSIONS. The monetary cost weight defines the user's willingness-to-pay for a higher connectivity performance. Our evaluation reveals that transmission time selection is of particular importance when the monetary cost is considered. If cost is neglected in the rating, i.e. the user does not care for the price to pay, most application QoS requirements can still be satisfied from currently available networks through flow prioritization and network selection. However, as soon as cost gains importance, moving delay-tolerant data transmissions to cheaper networks dominates the rating and renders pure network selection ineffective. Nevertheless, for any cost weight, our designed Joint Transmission Planning (JTP) continuously shows high NRS performance of in average 87.97%. If monetary cost is considered, it significantly outperforms the from state-of-the-art derived approaches in average by 21.00% (NS), 8.73% (ONS) and 11.44% (DWO).

4.4 SUMMARY AND CONCLUSIONS

In this chapter, we investigate the transmission quality of combined time-network selection, firstly, by designing a novel data transmission rating model and, secondly, by developing and evaluating a novel strategy for an explicit joint time-network selection that we implement in our approach Joint Transmission Planning (JTP).

Our novel transmission rating model, defining our first main contribution, covers application QoS requirements satisfaction and monetary cost. It integrates rating model components from transmission time selection and network selection and extends them with a new throughput requirement model, which is modeled according to the fundamental definition of throughput: an amount of data sent within a certain time span. The model parameterizes this time span to define by which degree a transmission is required to be continuous. If defined as a single time slot, the model enforces continuous and instant transmission, equivalently to models from network selection. Otherwise, the transmission is allowed to happen flexibly within the time span, enabling transfer in bursts. For a time span till the deadline, the model equals that of transmission time selection. Thus our novel throughput requirement model generalizes the two and is able to handle respective data transmissions in a unified way.

Consecutively, as our second main contribution, we designed our novel Joint Transmission Planning (JTP) with new design principles, integrating a sophisticated network selection into a transmission time planning. It selects the networks with the best flow-network matching within a certain planning time horizon and uses the time overlap between the prospective network availability and the desired transmission period of the data flow as a constraint for data allocation.

To evaluate JTP, we employ the designed rating model and compare JTP to predominant state-of-the-art strategies, each representing an upper performance bound for a certain transmission algorithm class. This is, firstly, a Network Selection (NS) that allocates data to the best-matching now-available networks. Secondly, we compare an Opportunistic Network Selection (ONS) that extends NS by the opportunity not to transmit data if the flow-network match is considered as insufficient. Thirdly, we employ a Delayed WiFi Offloading (DWO) that tries to move as much data transmissions as possible to prospective available WiFi networks without violating transmission deadlines.

In our evaluation, we use a fractional factorial evaluation design, varying (1) the scenario length, i.e. the planning time horizon in time slots, (2) the number of networks, (3) the amount of data to transmit, (4) the number of different data flows, i.e. data flow diversity and (5) the monetary cost weight, which balances the application QoS satisfaction component of the rating model with monetary cost. From this, we draw four main conclusions, ordered by their importance.

1. The novel Joint Transmission Planning (JTP) significantly outperforms the approaches derived from state-of-the-art in rated transmission performance by 7-26% and shows strong robustness against the applied parameter variations (except a reduced planning time horizon) reaching 87-91% of the scenario optimization potential.

2. The transmission performance of JTP is comparable to that when using Network Selection (NS) in an environment with 4 times more networks.

3. Network Selection strategies perform well compared to JTP in scenarios with (1) nearly no alternative networks, rendering selection at all obsolete, (2) a huge amount of alternative networks, providing excellent connectivity at nearly each point in time and rendering transmission time selection obsolete, (3) an exceptional high data traffic load, rendering selective transmission delaying ineffective because all available network resources are required for transmission and (4) ignoring monetary cost, as now-available networks can in most cases satisfy application QoS requirements using simple data flow prioritization, however, by employing predominantly expensive networks.

4. Transmission time selection strategies perform well compared to the transmission performance of JTP in scenarios with sparse network resources with some short-range networks within the planning time horizon, shifting delay-tolerant data traffic to time slots covering short-range networks, while keeping other resources free for time-critical data transfers.

In this chapter, we assumed the availability of complete information about networks and the data flows from a perfect prediction. In the next chapter, we investigate the impact of erroneous prediction and analyze its effect on the transmission planners. Consecutively, we design and evaluate an approach to treat arising issues.

5 TRANSMISSION PLAN ADAPTATION

This chapter treats the performance of transmission planning under erroneous prediction due to environmental changes and presents our approach to react to this kind of uncertainty. Long-Term transmission planning using joint time-network selection, as presented in the previous chapter, significantly improves the perceived performance of the Internet connection. However, it depends on predictive information. What happens if this prediction is erroneous? To investigate this question, we firstly develop prediction error models for the connected vehicle scenario that change essential characteristics of the predicted input values of transmission planning in a controlled manner, covering network characteristics, vehicle movement and data flows to transmit. Note that we do neither rely on real data nor investigate prediction methods. In contrast, we assume prediction to be available and analyze how prediction errors of different kinds and strengths affect the performance of our proposed models. From this, fundamental requirements for future prediction models can be derived. We present a transmission plan execution algorithm and show that its performance drops severely because it is unable to react to the environmental changes. This deficiency motivates for the design of more elaborated execution algorithms.

Secondly, we introduce our transmission plan adaptation as an extension to opportunistic transmission approaches, here applied to Opportunistic Transmission Planning (ONS). It implements the current transmission regarding an existing long-term plan while reacting immediately to environmental changes. To make the approach follow a transmission plan whenever it is feasible and, at the same time, to give freedom for adaptation when following the plan is considered ineffective, we design three mechanisms that control data allocation depending on the recognized environmental changes. Each of these mechanisms treats one certain prediction component: network characteristics, vehicle movement and data flows to transmit. We show that the resulting adaptation approach is able to sustain a significant share of the performance gain from long-term transmission planning and identify prediction error limits at which a re-planning should be preferred over an ongoing plan adaptation.

5.1 PREDICTION ERROR MODELS

To provide a potential reality for simulation of the transmission plan execution algorithms, we introduce prediction error models that derive actual values A_t from the predicted values P_t, which have been employed for transmission planning. The models implement a definable statistical prediction error between A_t and P_t, using the parameter γ to scale the strength for scenario randomization. In this section, we create such models for the three prediction components, network characteristics, vehicle movement and data flows to transmit.

To measure the error, we employ an extended Symmetrical Mean Absolute Percentage Error (SMAPE) [141], as defined in Equation 5.2. In contrast to the often used Mean Average Percentage Error (MAPE), SMAPE is additionally able to handle cases in which one of the two values is zero, without leading to infinite result values. In extension to the commonly used definition, we handle the case of no error explicitly $A_t = P_t$ in Equation 5.1, covering additionally the event of both values being zero. To define the error of the three prediction components, we select A_t and P_t individually for each model.

Prediction error models derive a possible reality with controlled statistical changes from a given prediction.

$$e(t) = \begin{cases} 0, & \text{if } A_t = P_t \\ \frac{|P_t - A_t|}{|P_t| + |A_t|} & \text{else} \end{cases} \tag{5.1}$$

$$\mathrm{SMAPE} = \frac{2}{T} \sum_{t=1}^{T} e(t) \tag{5.2}$$

5.1.1 *Network Characteristics Prediction Error*

The performance of an access network depends on the characteristics of its access points and the environment. Most parts of the environment are static, like the access point position, the landscape geometry or buildings. The signal propagation stays about the same in this environment. Moreover, network users follow similar patterns within their routines each day. These patterns allow network operators to plan provisioning of sufficient resources for the daily average use. Severe underprovision happens therefore mainly in locations, where the daily average need for network access is expected to be small and temporal peaks are neglected.

Factors affecting the perceived transmission performance beyond that can be categorized in predictable long-term and non-predictable short-term impact factors. A considerable share of the load is caused by heavy-tailed data traffic [187] middling out over time to a certain base load, which contributes to long-term impact factors, barely changing over several minutes. Furthermore, randomly occurring effects, like an unusual high user density, e.g. during public events or traffic jams, may create a temporally dominating impact on the perceived network characteristics due to network overload or even a transition [216, 216] of the underlying network mechanisms [176, 181]. Furthermore, there might be effects, as torrential rains, which affect the range of short-range network technologies. As these effects persist for at least several minutes, we also rate them as a long-term impact that can be considered in prediction but cause prediction errors during the transition process.

For these long-term effects, the perceived transmission quality can be mapped very well to the location, optionally varying in time patterns, using steadily updated connectivity maps [154, 166]. For such a map creation and distribution, advanced monitoring concepts [202, 178, 177] may be employed. Accordingly, average perceived network performance can be estimated well for the user location.

To model range-affecting effects, like in the torrential rains example, we employ a narrow and truncated Gaussian distribution to select a number of time slots to be

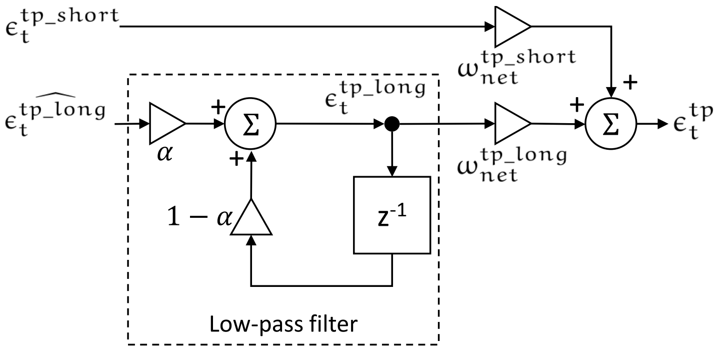


Figure 21: Network throughput prediction error model

removed from, respectively added to the center of the availability window of the short-term network. To model other throughput *prediction errors* caused by these long-term effects, we introduce a *target* offset $\epsilon_t^{\widehat{tp\_long}}(\gamma)$, which persists for a limited random time span and is selected from a truncated Gaussian distribution. The parameter γ scales the randomization strength, i.e. stretches or shrinks the standard deviation of the distribution. As we consider these long-term effects to change slowly, the actual long-term error $\epsilon_t^{tp\_long}$ of the model converges to this target offset $\epsilon_t^{\widehat{tp\_long}}(\gamma)$ over time, using a first order IIR low pass filter according to Equation 5.3.

In contrast, user interaction creates erratic load peaks at the networks. Partially, these peaks middle out, contributing to the base load. Nevertheless, superposing load peeks create barely predictable fluctuations, defining short-term impact on the network characteristics [92]. For our prediction error, we model a short-term throughput offset $\epsilon_t^{tp\_short}$ using a truncated Gaussian distribution, re-calculated for each single time slot (with length $\Delta\tau$ of, e.g., 1 second). It contributes to our network throughput prediction error ϵ_t^{tp}, as presented in Equation 5.4 and visualized in Figure 21. Since we defined latency and jitter characteristics as invariant within the limited planning time horizon, we vary those by a random offset from a truncated Gaussian distribution.

Throughput fluctuations from erratic network load superpose long-term prediction errors from a slowly changing environment.

Independently from the accuracy of the prediction itself, this model is able to simulate different magnitudes of prediction errors for later analysis. To determine the error strength SMAPE^{net} of a network, we calculate the weighted sum of the error values of throughput SMAPE^{tp}, latency SMAPE^{lcy} and jitter SMAPE^{jit}, according to Equation 5.5. In the evaluation in Section 5.3, we set them exemplary to $\omega_{net}^{lcy} = \omega_{net}^{jit} = 0.25$ and $\omega_{net}^{tp} = 0.5$. For each of them, we set A_t to the actual value and P_t to the predicted value. The error corresponds to the mean value over all networks.

$$\epsilon_t^{tp\_long} = \alpha \ \epsilon_t^{\widehat{tp\_long}}(\gamma) + (1-\alpha) \ \epsilon_{t-1}^{tp\_long} \tag{5.3}$$

$$\epsilon_t^{tp} = \omega_{net}^{tp\_long} \epsilon_t^{tp\_long} + \omega_{net}^{tp\_short} \ \epsilon_t^{tp\_short}(\gamma) \tag{5.4}$$

$$\mathrm{SMAPE}^{net} = \frac{1}{N}\sum_{n\in N} \omega^{tp}_{net}\ \mathrm{SMAPE}^{tp}_{n} + \omega^{lcy}_{net}\ \mathrm{SMAPE}^{lcy}_{n} + \omega^{jit}_{net}\ \mathrm{SMAPE}^{jit}_{n} \tag{5.5}$$

5.1.2 *Movement Prediction Error*

Movement of vehicles, and therefore the location at each point in time, depends, firstly, on the usual travel time on road segments and, secondly, on erratic road traffic events like traffic jams, red lights, overtaking maneuvers or crossing pedestrians. For arrival time estimation, employed for navigation, most of these effects may middle out over the trip time. However, especially for middle- and short-range networks, a spatial difference from the predicted location at a certain point in time may result in a substantially different perceived network environment because the vehicle is unexpectedly not in the covered area of a network.

Movement prediction errors shift, shrink or extend the time spans in which especially networks using short- and mid-range access points are available. For example, a slower moving vehicle will reach a network access point later than expected and stay longer in its covered range. The desired output for our movement prediction error model is the temporal offset for network availability.

To model the movement prediction error, we randomly select a target velocity $v_t^{target}(\gamma)$ from a truncated Gaussian distribution centered at the predicted velocity v_t^{pred} and derive a relative velocity target error $\epsilon_t^{\widehat{rel\_v}}$ for the vehicle in time slot t, as shown in Equation 5.6. The parameter γ scales the standard deviation of the Gaussian distribution, while its truncation stays fixed at speed zero and double the predicted speed. As velocity is differentiable, we smoothen the target velocity using a first order IIR low pass filter, as described in Equation 5.7 and depicted in Figure 22. Movement errors sum up over time. Therefore, we employ an integration to derive the temporal offset ϵ_t^{s} in time slots, describing the spatial error, presented in Equation 5.8.

Based on this offset, the vehicle perceives the network environment predicted to be available in another time slot. We can express this in time slot shifts referring to the initial plan, which we model in a compensation function that, in the case of a fast moving vehicle, may skip one time slot, or, in the case of a slow moving vehicle, keeps the network characteristics of a time slot for the next one. The model employs a compensation function $f_{comp}(x)$, presented in Equation 5.9, which adds or removes one time slot for compensation. Note that employing truncated distributions for movement randomization avoids errors beyond that, i.e. moving backward and moving faster than double the speed. As input to calculate the compensation ϵ_t^{comp} required in this time slot, the model uses the current temporal offset ϵ_t^{s} corrected by the sum of previous compensations Δt_t^{offset}, as shown in Equation 5.10. To derive the desired shifted network prediction as output, the movement error model looks up the initial time-slotted plan of network prediction with the calculated time offset Δt_t^{offset} from Equation 5.11.

Movement prediction errors modify the temporal availability of predicted networks, originating from differences in the predicted vehicle velocity and the resulting vehicle location over time.

To measure the error strength, we define SMAPE^{move} using $A_t = 0$ and $P_t = \epsilon_t^{comp}/T$, resulting in Equation 5.12. Setting $A_t = 0$ means that an accurate prediction does not include any compensation. However, each compensation to any time

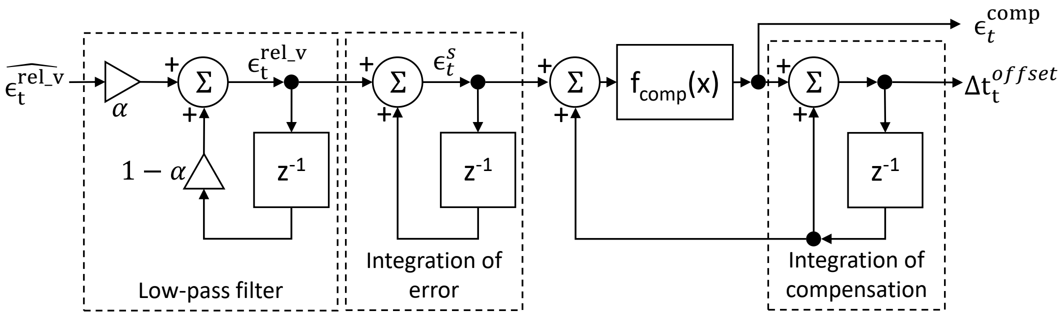


Figure 22: Movement prediction error model

slot t contributes to the error, normalized to the predicted time span T. An average required compensation of 1 in every time slot results in a maximum SMAPE prediction error, reflecting the extreme cases of standing still or moving with at least double speed.

$$\widehat{\epsilon_t^{rel\_v}} = \frac{v_t^{target}(\gamma) - v_t^{pred}}{v_t^{pred}} = \frac{v_t^{target}(\gamma)}{v_t^{pred}} - 1 \tag{5.6}$$

$$\epsilon_t^{rel\_v} = \alpha\ \widehat{\epsilon_t^{rel\_v}} + (1-\alpha)\ \epsilon_{t-1}^{rel\_v} \tag{5.7}$$

$$\epsilon_t^s = \epsilon_t^{rel\_v} + \epsilon_{t-1}^s \tag{5.8}$$

$$f_{comp}(x) = \begin{cases} 1 & \text{if } x < -0.5 \\ -1 & \text{if } x > 0.5 \\ 0 & else \end{cases} \tag{5.9}$$

$$\Delta\epsilon_t^{comp} = f_{comp}(\epsilon_t^s + \Delta t_{t-1}^{offset}) \tag{5.10}$$

$$\Delta t_t^{offset} = \epsilon_t^{comp} + \Delta t_{t-1}^{offset} \tag{5.11}$$

$$SMAPE^{move} = \frac{1}{T} \cdot \sum_{t=1}^{T} |\epsilon^{comp}| \tag{5.12}$$

5.1.3 *Data Flow Prediction Error*

Data flow prediction errors originate from unexpected events, triggering data transfer in the background or from user interaction. The user may initiate, delay or cancel a data transfer. Accordingly, we model data flow prediction errors in three consecutive steps, reflecting those three actions.

Data flow prediction errors add, cancel or pause transmissions, caused by unexpected events and user interaction.

Firstly, we add F data flows, each with a probability of p_{add}. Added data flows start at a random time slot within the planning time horizon. Their types are preferred in the following order: interactive, bufferable stream, conversational, download. The flows follow randomized but typical characteristics of these categories, c.f. Appendix A.3. The exact deviation depends on the desired scenario and user. We provide selected values in the evaluation. Secondly, we cancel data flows with the same probability p_{cancel} for each flow from the new extended set. Cancellation happens at a random time step from a uniform distribution with a peak at zero duration for complete flow cancellation. In the third step, the model selects canceled flows to be continued at a later point in time, which corresponds to an intended pause or delay for the data transfer. The paused data flows continue with same requirements and the remaining number of data tokens, as well as a shifted deadline. For data flow prediction errors, the parameter γ scales the probability with which data flows are modified.

To calculate the SMAPE for data flows $SMAPE^{flow}$, we set for each data flow $A_{t,f} = 1$ when the transmission is desired in the current time slot and $P_{t,f} = 1$ if the desired transmission is predicted in time slot t, c.f. Equations 5.13 and 5.14. Hence, it rates the overlap of the active times between the start time and the deadline for each data flow in each time slot, reducing other parameters to the temporal overlap. We do not consider other parameters, as a substantial change in other data flow requirements is very similar to a removing the previous data flow and adding a new one, which is covered in our model. Finally, according to Equation 5.15, we measure the mean SMAPE over all considered data flows.

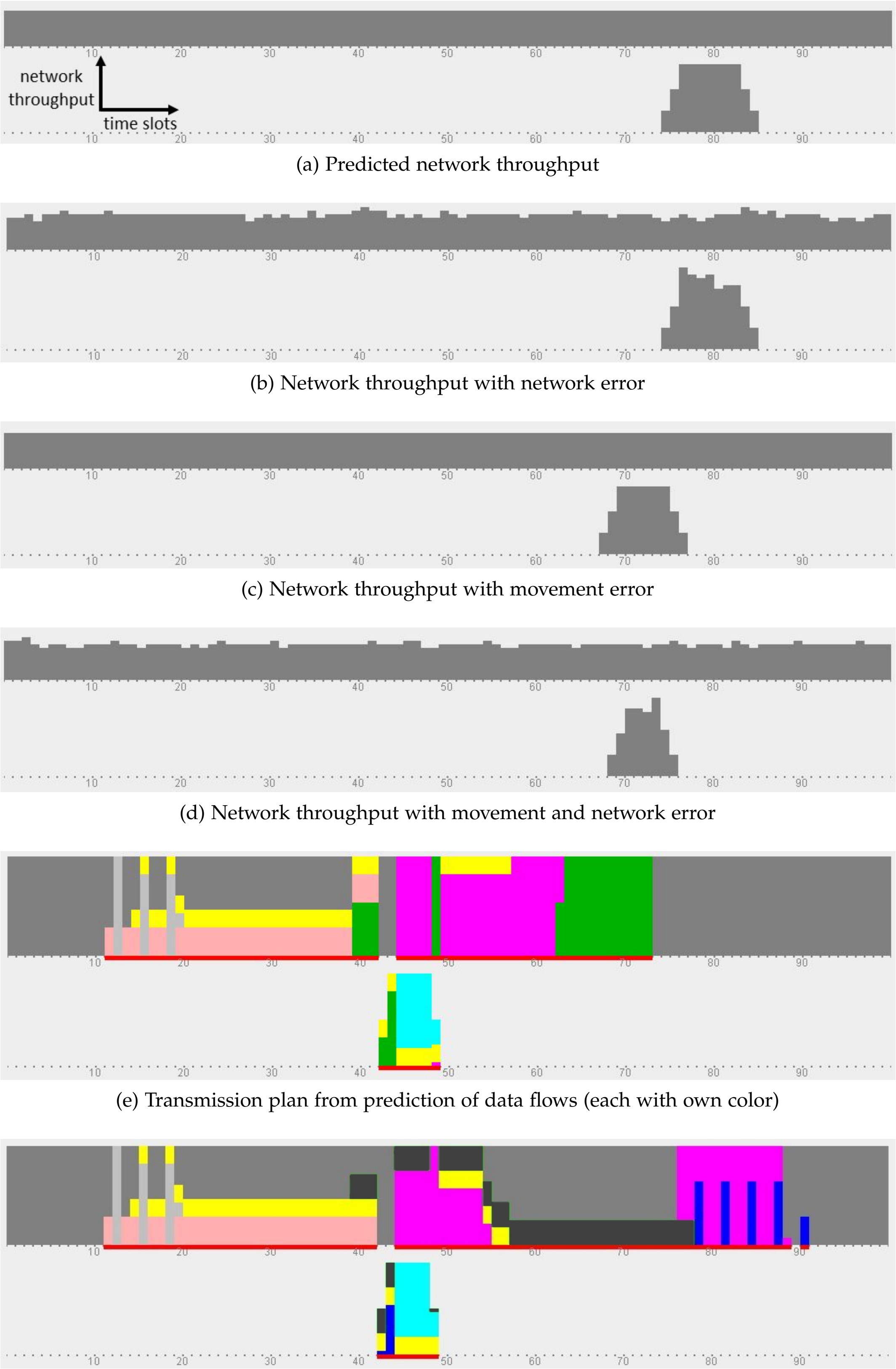
$$A_{t,f} = \begin{cases} 1, & \text{if } t^{st}_{f,actual} \leqslant t \leqslant t^{dl}_{f,actual} \\ 0, & \text{else} \end{cases} \tag{5.13}$$

$$P_{t,f} = \begin{cases} 1, & \text{if } t^{st}_{f,pred} \leqslant t \leqslant t^{dl}_{f,pred} \\ 0, & \text{else} \end{cases} \tag{5.14}$$

$$SMAPE^{flow} = \frac{1}{T \cdot F} \sum_{\substack{t \in T \\ f \in F}} |A_{t,f} - P_{t,f}| \tag{5.15}$$

5.1.4 *Prediction Error Examples*

Figure 23 gives examples for the prediction error types. Figure 23 (a) shows a predicted network environment, covering two networks, illustrated as dark gray



(a) Predicted network throughput

(b) Network throughput with network error

(c) Network throughput with movement error

(d) Network throughput with movement and network error

(e) Transmission plan from prediction of data flows (each with own color)

(f) Executed transmission plan of data flows with data flow prediction error

Figure 23: Throughput for two networks over time. Network, movement and flow prediction error examples with SMAPE strength 0.2.

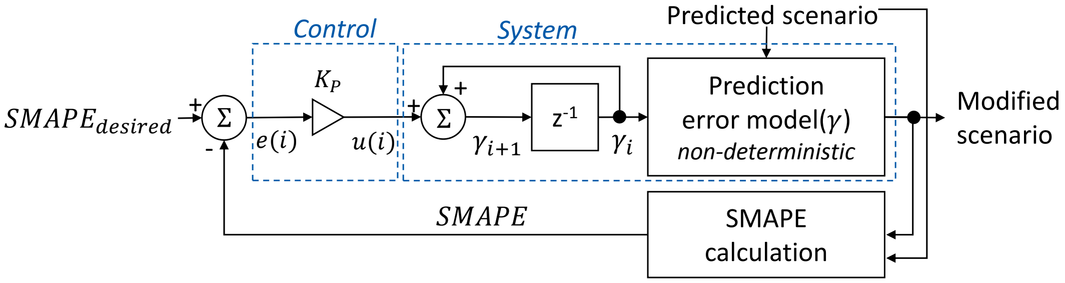


Figure 24: Prediction error adjustment using a feedback control loop

shapes. The x-axis denotes time, whereas the y-axis shows the maximum throughput over time, which a node can expect from this network. In the example, this may represent a cellular network on the top, providing about constant throughput, and a passed-by WiFi network which is available for 11 time slots, e.g. 22 seconds at planning granularity of $\Delta\tau = 2s$. Figure 23 (b) demonstrates the effect of the network prediction error of 0.2. The perceived network throughput differs slightly from the predicted values in every time slot. The impact of a movement prediction error is illustrated in Figure 23 (c), showing the case of a vehicle moving faster than expected. The effect is not visible on the exemplified cellular network because it provides constant throughput. In contrast, the WiFi, whose availability is limited to a small area, gets into reach earlier and is available for one time slot less because the vehicle passes it faster than expected. The two error models are combined in 23 (d), keeping the movement prediction error. In addition, the range of the WiFi network is decreased, which might be caused by environmental effects, torrential rains.

The two remaining Figures 23 (e) and (f) show the effect of the data flow prediction error on an example transmission plan, consisting of allocated data to networks and time. Each color represents a data flow, which is allocated to networks at corresponding time slots with a certain throughput, encoded by the height of the bars. The transmission plan of Figure 23 (e) is derived from a prediction of 6 data flows. In the example, the prediction error leads to the following changes: (1) the dark green data flow is canceled completely (2) two new data flows are added, the anthracite and the blue one (3) the magenta data flow is paused and continued later.

5.1.5 *Prediction Error Adjustment*

To evaluate the prediction error resistance of our algorithms, we have to derive modified scenarios with a defined $\text{SMAPE}_{\text{desired}}$ error to the initial scenario, representing the prediction. For all three prediction error models, the error strength can be influenced using the parameter γ. It scales trigger probabilities and the standard deviations of distributions used for randomization. However, since the models apply randomized modifications to the predicted scenario, the modifications and their strength (in SMAPE) are non-deterministic. Hence, the resulting SMAPE is neither deterministic nor proportional to γ but statistically correlates

Table 12: Control parameters for the prediction error model creation feedback control

	Network	Movement	Data Flow
K_P	0.45	0.09	0.03

positively with it. Based on this correlation, we assume that the system has a linear behavior. Hence, the control loop can then be developed with conventional linear system theory, employing a P-control, which multiplies the control error $e(i) = SMAPE_{desired} - SMAPE$ with a factor K_P and add the product to γ in order to receive γ_{i+1} of the next iteration i. We continue updating the control error in consecutive iterations as long as it exceeds e_{stop}, which we set to 0.01 for later evaluation.

To tune the feedback control parameter K_P empirically, we employ the Ziegler-Nichols method [225]. Accordingly, we increased the parameter K_P slowly over multiple runs until γ_i starts to oscillate, defining K_{crit}. Finally, we set $K_P = 0.5\ K_{crit}$. As long as the controlled error does not diverge over iterations due to controlled loop instability, a non-optimal parameter tuning is not critical. It leads in the worst case to a slower convergence towards $SMAPE_{desired}$ in the feedback control loop and therefore to additional CPU time required for the scenario creation process, which is relevant for simulation setup only. Therefore, convergence and stability have not been investigated in detail. In tests, the result usually converges to the desired result within a few iterations.

$$\gamma_{i+1} = K_P(\underbrace{SMAPE_{desired} - SMAPE}_{\text{control error } e(i)}) + \gamma_i \tag{5.16}$$

5.2 TRANSMISSION PLAN ADAPTATION ALGORITHM

In this section, we design a transmission plan adaptation targeting robustness of transmission quality even under erroneous prediction. We first design a transmission plan execution algorithm Exec(p) that is able to follow a given plan p. Analysis of the execution reveals that it fails to sustain the performance gain from planning in the presence of even small prediction errors due to its inability to react to environmental changes. It leads to a severe drop in executed transmission plan quality. In contrast, online algorithms, like Opportunistic Network Selection (ONS), improve only the current state without considering prediction. They do not integrate the time selection explicitly and, therefore, are not able to exploit the full optimization potential of the analyzed transmission optimization problem. To receive an algorithm that is flexible enough to react to environmental changes and that also integrates time dimension, we combine the two concepts in this section. Therefore, we use the results of long-term transmission planning to make the execution smart as well, employing opportunistic transmission to react to environmental changes. We call this transmission plan adaptation Ada(p).

5.2.1 *Adaptation Concept*

The adaptation algorithm is based on an execution Exec(p) derived from an opportunistic transmission approach and touches only the current time slot of the long-term communication plan. To let the execution follow the plan, we constrain the decision of the underlying opportunistic approach whether to transmit data or not, fostering each transmission that complies with the plan and suppressing each transmission that does not comply with the plan.

The adaptation approach selectively relaxes constraints of an execution with an underlying opportunistic approach in order to allow opportunistic reaction to environmental changes.

Analyzing detected errors of the prediction on which the plan is based, the adaptation approach selectively relaxes the introduced constraints of execution and adapts parameters, unlocking selected transmission opportunities beyond the plan as a reaction to environmental changes. For each kind of prediction error, network characteristics, node movement and data to transmit, we present one dedicated mechanism, which relaxes constraints if the detected environmental changes are expected to impair the transmission quality, implementing an opportunistic plan adaptation.

In the following, we first detail the execution algorithm, which extends an opportunistic approach with additional constraints. Second, we present the three mechanisms relaxing the transmission constraints, enabling opportunistic adaptation of transmission plans when considered beneficial.

5.2.2 *Plan Execution Algorithm*

To explain the execution algorithm, we first recover the basics of transmission rating from Chapter 4 and, based on that, explain how execution extends opportunistic approaches to follow the plan.

The transmission rating is based on a *model of forces*, presented in 4.1, briefly recovered in the following. The rating function sums up two components, which are active in a mutually exclusive manner, c.f. Equation 4.5. The first component, the repelling forces $c^{rep}(p_{f,t,n})$, is active when data is allocated. They punish the violation of data flow requirements from allocation to networks with insufficient transmission characteristics. As cost is to be minimized, the repelling forces let data push itself away from non-matching networks. The second component, the attracting forces $c^{attr}(p^*_{f,t})$, is active when data is *not* allocated. Note that $p^* = p$ is a visual aid to indicate that the employed cost function component creates a cost for non-allocated tokens. They punish dropped data and violation of minimum throughput requirements, pushing non-allocated data towards networks.

Opportunistic approaches, as the presented Opportunistic Network Selection (ONS) in Section 4.2.1, allocate data only if an estimated cost-benefit value, here defined as the difference between estimated attracting $c^{\widetilde{attr}}(p^*_{f,t})$ and estimated repelling forces $c^{\widetilde{rep}}(p_{f,t,n})$, exceeds the cost threshold c_{lim}, as presented in Equation 5.17. To make the underlying opportunistic approach follow the transmission plan, the execution algorithm constraints the opportunistic approach dynamically due to adapting the threshold c_{lim}. If transmission should be suppressed according to the given plan, it increases the threshold c_{lim} to a maximum value $c_{max}(f)$, defined by the supremum of the attracting forces according to 5.18. In contrast, for desired transmissions, the threshold c_{lim} is set to a minimum value c_{min}, defined as the infimum of the repelling forces according to 5.19.

For the decision about whether to transmit or suppress allocation, i.e. which of the two limits should be applied, the algorithm relies on a comparison of planned and already allocated data for a data flow f to a network n at the current point in time t_0, as shown in Equation 5.22. The algorithm releases the amount of planned data $p_{f,t_0,n}$ for transmission, according to Equation 5.20, which is defined by the plan p and does not vary during the current time slot t_0. In contrast, the amount of already allocated data $s_{f,t_0,n}$ in the current time slot t_0, shown in Equation 5.21, is continuously updated. As a result, the mechanism from Equation 5.22 fosters planned transmission and, as soon as the amount of planned data has been transmitted via the network in the current time slot, it suppresses transmission completely. With a perfect prediction, this results in implementing the plan accurately.

The execution constraints the decision of an underlying opportunistic approach whether to transmit or not to make it follow a given transmission plan.

$$\text{allocate if} \quad \widetilde{c^{attr}}(p^*_{f,t}) - \widetilde{c^{rep}}(p_{f,t,n}) > c_{lim} \tag{5.17}$$

$$c_{max}(f,t) = \sup_{t \in T} \widetilde{c^{attr}}(p^*_{f,t}) \tag{5.18}$$

$$c_{min}(f,t) = \inf_{n \in N_0} \widetilde{c^{rep}}(p_{f,t,n}) \tag{5.19}$$

$$p^{rel}(f,t_0,n) = p_{f,t_0,n} \tag{5.20}$$

$$s^{alloc}(f,t_0,n) = s_{f,t_0,n} \tag{5.21}$$

$$c_{lim} = \begin{cases} c_{min}(f), & s^{alloc}(f,t_0,n) < p^{rel}(f,t_0,n) \\ c_{max}(f), & else \end{cases} \tag{5.22}$$

The execution algorithm Exec(p) performs well for perfect prediction but fails to deliver acceptable results in the case of prediction errors, as presented in the evaluation in Section 5.3, because it is not able to react to environmental changes. The underlying opportunistic approach, in contrast, is fully reactive and does not rely on prediction. The following three adaptation mechanisms relax the constraints and modify parameters of the execution algorithm in a way that, in the case of prediction errors, it transmits data partially opportunistically, preserving the time-network selection from the plan if possible. Each of the adaptation mechanisms handles a certain kind of prediction error, network characteristics, node movement and data flow changes.

5.2.3 *Extended Data Release Mechanism*

The extended data release mechanism allows the underlying opportunistic approach to transmit data as soon as its transmission is planned and also later. The transmission benefit is re-evaluated.

The *extended data release* Mechanism is dedicated to handling network changes, relaxing, firstly, the network selection restrictions from the execution algorithm and, secondly, the data release limitations of the algorithm. Changed network characteristics can even affect whether it is beneficial to transmit or not. The opportunistic approach is able to handle exactly this decision on its own by definition. Hence, to relax the compliance to the planning and pass this responsibility, we modify the lower threshold limit c_{min}, setting it to zero according to equation 5.23, which is the default threshold of the opportunistic approach ONS. In this case, the approach decides for transmission as soon as it is considered beneficial, i.e. when the estimated attracting forces exceed the repelling ones, defined by the rating function.

$$c_{min}(f,t) = 0 \tag{5.23}$$

Furthermore, we handle network characteristic changes influencing the network prioritization. In this case, we ignore the planned network selection and consider planned time selection only. Therefore, the adaptation mechanism stops distinguishing between networks, considering the sum of allocated data in the current time slot t_0 over all networks N instead of checking each network individually. This constraint relaxation leads to an automatic fallback to the opportunistic approach's network selection algorithm. This is reflected in the sum over networks in Equations 5.24 and 5.25.

Finally, network characteristics might cover unbiased short-term fluctuations of the network throughput, based on the network load due to other clients, which are close to impossible to foresee and, thus, usually averaged out for prediction. Hence, the mechanism relaxes the amount of released data from the current time slot, releasing data for each time slots after the planned transmission. In the case of a lower data rate in the current time slot, not all data planned for allocation can be transmitted. In the opposite case of additional data resources being available, the mechanism fills gaps with data, which has been allocated in the plan earlier but could not be transmitted, plus data, which has not been allocated in the plan p_f^*, according to the Equations 5.24. and 5.25. The new mechanism allows to transmitting this data at a later point in time when there exist additional free network resources. Furthermore, data, which was not planned or failed to be transmitted, is used to fill free additional network resources if considered beneficial.

$$p^{rel}(f,t_0,n) = p_f^* + \sum_{t=0}^{t_0} \sum_{n \in N} p_{f,t,n} \tag{5.24}$$

$$s^{alloc}(f,t_0,n) = \sum_{t=0}^{t_0} \sum_{n \in N} s_{f,t,n} \tag{5.25}$$

5.2.4 *Location Reference Mechanism*

To cope with movement prediction errors, we present our corresponding adaptation mechanism, which refers to the initial plan by vehicle location instead of time. When a vehicle moves e.g. faster than predicted, it reaches and leaves short range networks earlier than expected, as shown in the example in Figure 23 (a,c). Compared to the prediction, location-dependent network characteristics move to another point in time. As a result, network availability is modified from the initial time-line, impacting on the network selection of the transmission plan. However, for delay-tolerant data transfers, the impact of network selection according to the plan dominates the impact of allocating data at the planned transmission time. To handle this issue, we introduce our *location reference* mechanism.

The location reference mechanism moves transmission in time, if not violating data flow requirements, to sustain the initially planned network selection.

For delay-tolerant data, it refers to the time slot in the transmission plan, which corresponds to the current vehicle location, preserving the initial network selection. Thus it considers the spatial dimension of the plan, ignoring the temporal one, employing a temporal offset $\epsilon_{move}(t_0)$ to refer to the plan.

However, for non-delay tolerant data flows, e.g. interactive ones, this temporal offset for transmission may lead to a violation of temporal QoS requirements, like throughput continuity or a deadline. Hence, we limit the temporal offset $\epsilon_{move}(t_0)$ to the maximum delay-tolerance of the data flow, which our model from Section 4.1.2.4 encodes in a throughput requirement window parameter $\Delta\widehat{t}_f^{min}$, according to Equation 5.26. To determine the location referenced time slot $t^{loc}(f, t_0)$, we add or remove the offset from the current time slot t_0 according to Equation 5.27, depending on whether the temporal offset $\epsilon_{move}(t_0)$ is positive or negative.

$$t_f^{offset} = \min(\Delta\widehat{t}_f^{min}, \|\epsilon_{move}(t_0)\|) \tag{5.26}$$

$$t^{loc}(f, t_0) = \begin{cases} t_0 + t_f^{offset}, & \epsilon_{move}(t_0) > 0 \\ t_0 - t_f^{offset}, & else \end{cases} \tag{5.27}$$

Referring to the correspondent vehicle location to release data covers a substantial issue: whenever the car stops, there is no further change of the location and, thus, no additional data is released for transmission. The transmission pauses even though there might be sufficient resources. The issue arises similarly when the vehicle moves slower than expected. To address this problem, the mechanism relaxes the condition for selecting the opportunistic transmission threshold c_{lim}. It releases data whenever a transmission for the flow is planned in the referred to time slot, as shown in Equation 5.28, replacing the trigger used in Equation 5.22. Hence, it is an additional trigger to allow transmission.

$$s^{alloc}(f, t_0, n) < p^{rel}(f, t^{loc}(f, t_0), n) \quad \text{or} \quad \left(\sum_{n \in N} p_{f, t^{loc}(f, t_0), n}\right) > 0 \tag{5.28}$$

The transmission condition acts independently from the condition of the *extended release* mechanism. Therefore, data allocated from this condition may not be released yet. Hence, it 'borrows' unreleased data to transmit from consecutive time slots in which their transmission was planned. Hence, adding robustness against slow vehicle movement, it completes our *location reference* mechanism. Accordingly, the *location reference* mechanism is designed to handle especially movement prediction errors. It, firstly, preserves network selection for delay-tolerant data and, secondly, handles data release shortages during a slower than expected vehicle movement.

5.2.5 *Flow Prediction Error Handling*

The third mechanism of our adaptation algorithm handles flow prediction errors. Flow prediction errors refer to additional data to be transmitted, time shifts in data transmission and canceled data transmission. For the first case of new data, there exist no reference in the plan. Therefore, we release this data instantly for opportunistic transmission, handling it equivalently to the non-allocated data p_f^* in the plan. This way, the opportunistic algorithm automatically prioritizes active data flows, integrating the new ones into the ongoing transmission. In the case of canceled data transmission, other data flows may consume the freed up additional network resources, which is similar to paused and moved data transmissions. This behavior is covered for already active transmissions from the *location reference* mechanism. For all other data flows, this is handled of the underlying opportunistic approach as follows. We use the SMAPE flow prediction error $\mathrm{SMAPE}^{flow}(f,t)$ to determine the change of the data flow. For small changes at a dedicated flow f, the planning might still be efficient. However, with a rising error for the flow, the probability rises that the planning is inefficient. Hence, we design a smooth transformation from the planned to an opportunistic transmission in order to react to flow prediction errors. By default, the execution mechanism sets the opportunistic threshold c_{lim} to c_{max} to suppress a transmission of a data flow. In contrast, when flow prediction errors occur, our approach does not suppress data allocation but restrict it to opportunities in which at least an error-dependent benefit threshold is reached. Therefore, we multiply the value c_{lim}^{max} with a function $\alpha(f,t) \in [0..1]$ according to Equation 5.31. In our model, we select $\alpha(f,t)$ according to Equation 5.29 depending on the SMAPE flow error function between the current point in time t_0 and one throughput window length $\Delta\hat{t}_f^{min}$ back. Why do we limit this time span? The flow prediction error, as defined in Section 5.1.3, calculates the mean of the error over the analyzed time slots. Applying the mean value equals a low pass filter and suppresses dynamics in the result. Accordingly, if the analyzed time span were not limited but covered the entire planning time horizon, the filter length would rise as well. Therefore, the reaction to flow prediction errors at the very start of the prediction time horizon would be heavier than at a later point in time, because a single value takes less effect in the mean. When a higher number of other values is considered, it is essential to limit the analyzed time span for flow prediction errors. Why do we select the throughput window length $\Delta\hat{t}_f^{min}$ of the data flow? As discussed above, $\Delta\hat{t}_f^{min}$ indicates the delay-tolerance of a data flow. Hence, a data flow with a short delay tolerance requires fast reaction to detected flow prediction errors. Keeping its desired transmission time is more critical than keeping a planned network selection. In contrast, a slight shift of e.g. the desired

Flow prediction error handling enables transmission of new data flows, weaving them into the transmission opportunistically. In the case of prediction errors, it additionally unlocks beneficial transmission opportunities beyond the plan.

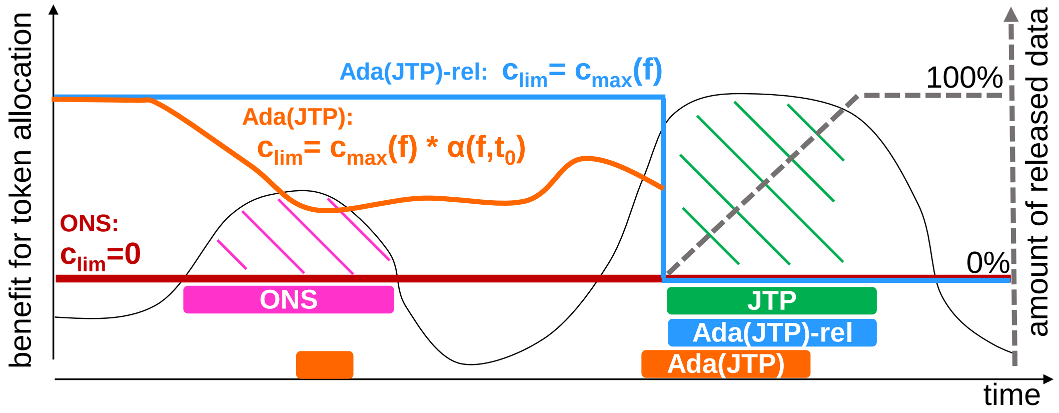

benefit for token allocation
Ada(JTP)-rel: $c_{lim}= c_{max}(f)$
Ada(JTP):
$c_{lim}= c_{max}(f) * \alpha(f,t_0)$
ONS:
$c_{lim}=0$
100%
0%
amount of released data
ONS
JTP
Ada(JTP)-rel
Ada(JTP)
time

Figure 25: Schematic of adaptation algorithm: Basic data release mechanism (blue) and flow prediction error handling (orange).

transmission start for a highly delay-tolerant data transfer might not influence the feasibility of a transmission plan at all.

In fact, the function $\alpha(f,t)$ decreases the strength of the suppression of unreleased data, depending on the flow prediction error. Reducing c_{lim}^{max} allows the underlying opportunistic approach to allocate unreleased data to well matching opportunities. Thus, at an entirely wrong prediction, the algorithm behavior converges to a fully opportunistic transmission of the data flow. A slightly wrong prediction enables early opportunistic transmissions, whenever opportunities come up for which a sufficiently high benefit is expected. Thus, prediction errors degrade the threshold of the underlying opportunistic approach. As a result, the transmission may be earlier, i.e. more conservative, as it is not restricted to the best-expected opportunity. The finally resulting adaptation algorithm threshold is given in Equation 5.31.

$$\alpha(f,t_0) = 1 - \sum_{t=t_0-\Delta\hat{t}_f^{min}}^{t_0} \frac{\epsilon_{flow}(f,t)}{\Delta\hat{t}_f^{min}} \tag{5.29}$$

$$c_{max}(f,t_0) = \sup_{t\in T} c_{attr}(p^*_{f,t,n}) \cdot \alpha(f,t_0) \tag{5.30}$$

Final adaptation algorithm threshold:

$$c_{lim} = \begin{cases} 0, & s^{alloc}(f,t_0,n) < p^{rel}(f,t^{loc}(f,t_0),n) \\ & \text{or} \quad \left(\sum_{n\in N} p_{f,t^{loc}(f,t_0),n}\right) > 0 \\ \sup_{t\in T} c_{attr}(p^*_{f,t,n}) \cdot \alpha(f,t_0), & \text{else} \end{cases} \tag{5.31}$$

5.2.6 *Adaptation Design Discussion*

To illustrate the characteristics of our adaptation mechanism, we give an example in Figure 25. It shows the maximum benefit for a data allocation over time as a thin black curve. In the example, it has two maxima. In addition, the figure shows the default cost-benefit threshold for allocation decision of Opportunistic Network Selection (ONS) $c_{\mathtt{lim}} = 0$ as a red horizontal line over time. Opportunistic Network Selection transmits data at the earliest opportunity for which it expects a benefit, here during the left 'hill'. In contrast, Joint Transmission Planning (JTP) analyzes the opportunities within the entire planning time horizon and selects the best-expected ones for transmission. In the example, this corresponds to the right and higher benefit 'hill'. The transmission time spans of the approaches are marked as colored bars in the figure. To visualize of the data release mechanism, we draw a gray dashed line at the right, which exemplifies the amount of released data in the plan of JTP for a data flow over time.

With perfect prediction, the adaptation mechanism releases data, which the underlying opportunistic approach instantly allocates to the according networks. As a result of newly released data in these time slots, the opportunistic transmission threshold $c_{\mathtt{lim}}$ is set to the ONS default zero, as illustrated with the blue line of Ada(JTP)-rel. The blue bar below it indicates similar transmission of the data flow as planned. What happens in the the case of erroneous predictions? In the case of a data flow prediction error, the α-function reduces the threshold $c_{\mathtt{lim}}$ dynamically, illustrated in the figure as an orange curved line, Ada(JTP). In the example, this leads to the partial earlier and more conservative transmission of the data flow, indicated by the orange bar. In the following, we present four selected prediction error scenarios and explain the adaptation approach's automatic reactions.

Network latency We consider a case in which the latency of the network planned for allocation is substantially higher than expected, leading to an application requirement violation during strict execution. In contrast, the adaptation Ada(p) releases data for allocation but does neither enforce to follow the network selection nor the allocation decision. After analyzing the actual environment, the underlying opportunistic approach decides whether another network suits better for transmission or not.

Throughput shortage In the case of lower throughput than expected for the target network, not all released data can be transmitted as soon as planned. The release mechanism allows the approach to continue the data transfer beyond the planned transmission time's end. The same holds for allocation of additional data to be transmitted, which causes the same effect as data with an earlier expected transmission. The underlying opportunistic approach automatically prioritizes data flows and decides whether to reduce data rates of flows or to take over resources, which have been reserved for another data flow in the planning. This behavior can cope with resource shortages efficiently, using the opportunistic fallback strategy to weave in unplanned data.

Throughput over-provision

During unexpected high throughput phases or a cancellation of a planned data transmission, a straight plan execution cannot use the additional free resources. In contrast, the *location reference* mechanism enables transmission of as many tokens as possible whenever a data transmission is planned. These additional allocations can speed up the data transfer beyond the planning. Furthermore, the underlying opportunistic approach automatically fills up the unused resources with new, dropped or delayed data of other flows if this is considered beneficial.

Different Movement

Faster or slower movement of the vehicle than expected leads to a location offset, changing the perceived network environment at a certain point in time. Since planning selects the transmission time with respect to the availability of a matching network, referring to transmission time does not completely reflect the intent of the planning. Networks of the planned transmission might be unavailable. Thus, a straight execution fails. In contrast, our location reference mechanism of the adaptation preserves the network selection by re-selection of the transmission time in the limits of the flow's delay-tolerance. Therefore, it looks up the transmission plan at a reference time slot corresponding to the current vehicle location. This re-selection of time might lead to a significantly new distribution of data because the transmissions of different data flows are moved independently from each other regarding their individually limited time offsets. The underlying opportunistic approach implements a possible re-allocation for this new data distribution.

Conclusively, the three mechanisms of our adaptation algorithm sustain beneficial characteristics of planning whenever possible. In particular, they cope with all three identified kinds of prediction errors: network characteristics, vehicle movement and data flow prediction errors. Furthermore, the underlying opportunistic approach for adaptation guarantees feasibility of the data allocation and changes planned decisions on the fly when more beneficial transmission opportunities arise. Thus, our adaptation algorithm combines the advantages of the two worlds: the flexibility of opportunistic approaches and the long-term benefits from planning.

Our adaptation algorithm Ada(p) is integrated into the data allocation decision of opportunistic network selection approaches. The underlying opportunistic approach of our adaptation algorithm is, in general, exchangeable. For evaluation, we apply the algorithm to the Opportunistic Network Selection (ONS), which applies the heuristics developed in Chapter 4.2.2. Its core is the basic Network Selection (NS), which it extends to the opportunistic allocation decision. To use it as adaptation, the approach integrates our above-designed mechanisms. Therefore, it replaces the constant c_{lim} by the dynamic adaptation component in Equation 5.31.

5.3 EVALUATION

In this section, we evaluate the resulting transmission quality and the execution time of the adaptation approach and compare it to pure transmission plan execution and the online approach ONS. Firstly, we detail the evaluation setup with controlled and dependent variables and, secondly, we present and discuss the results.

5.3.1 *Evaluation Setup*

To evaluate the adaptation, we measure its performance in scenarios with prediction errors (SMAPE) as controlled variable and compare it to that of the opportunistic approach (ONS) and, for reference, to that of the Joint Transmission Planning (JTP) with perfect prediction. To rate the performance, we apply the Normalized Rating Score (NRS), according to 4.3.1.1. It normalizes the transmission performance, defined in our rating model in Section 4.3.1, between the cost function value of an optimal plan and the average of random transmission plans and determines the used share of the scenario's optimization potential, defined as the margin between these two bounds. We vary the SMAPE prediction error between 0.0 and 0.5, isolated for each of the three kinds of prediction error. Finally, we present a combined prediction error, which superposes the three error models in one scenario with the same SMAPE for each of the three. In addition, we provide an analysis of the algorithm execution times. Time measurements have been performed using a single core of a server machine with Intel Xeon E5-2643 v3 @ 3.4GHz and 512 GB RAM. For each run, we evaluated 50 randomized scenarios, each with 100 time slots planning horizon length, 8 data flows and 8 networks, equivalently to the default evaluation setup in Chapter 4. Note that the number of data flows may vary in the flow and combined prediction error evaluation due to adding or canceling data flows, according to the model. To show the typical performance and its distribution, we give the $Q_{25\%}$, $Q_{50\%}$ (median) and $Q_{75\%}$ quantiles.

We show the performance of the adaptation approach with underlying ONS, executing transmission plans of Joint Transmission Planning approach (JTP) and activate different parts of the adaptation mechanism to show their effects. Exec(JTP) is a pure execution of the transmission plan. The Ada(JTP)-rel represents the adaptation approach with only the extended release mechanism activated, aiming to provide robustness against network changes. Ada(JTP)-rel-loc provides the results of the adaptation after adding the *location reference* mechanism, addressing movement prediction errors. The final approach, incorporating all three mechanisms, also the *flow prediction error handling*, is denoted by Ada(JTP). Furthermore, we compare the results of applying adaptation to an optimal plan to estimate the benefits of using advanced planning approaches.

5.3.2 *Impact of Network Prediction Errors*

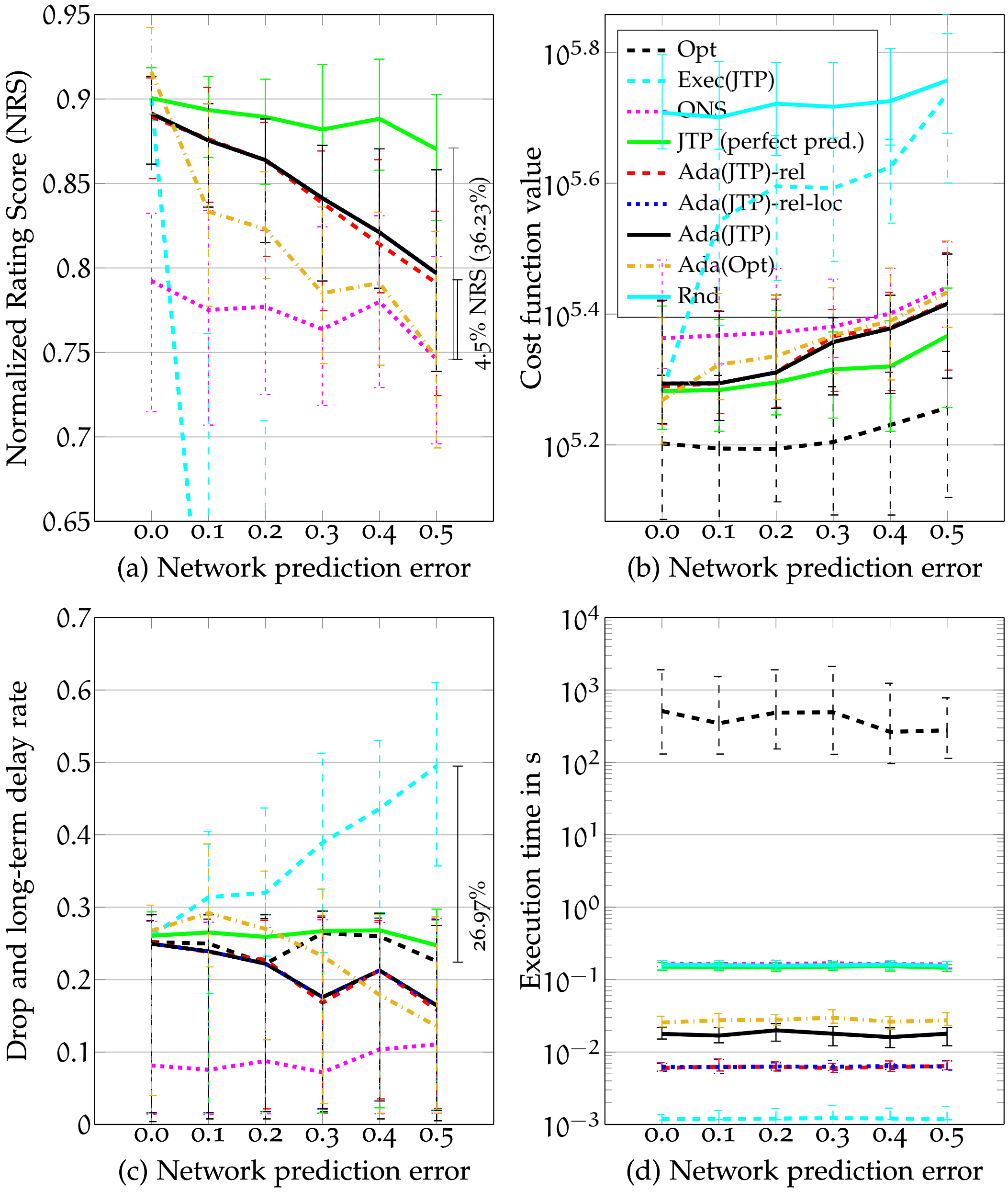
The SMAPE network prediction error quantifies the changes in the network throughput, latency and jitter compared to the prediction. Those changes can reduce or increase the amount of transmittable data and affect the flow-network matching. The error model is dominated by short-term network throughput fluctuations, resulting from the activity of other users sharing network resources. We analyze the effects of this error on execution Exec(JTP) and adaptation Ada(JTP) of plans from Joint Transmission Planning (JTP) as well as optimal plans Ada(Opt).

PLAN EXECUTION. Figure 26 (a) shows the Normalized Rating Score (NRS) of the analyzed approaches. Even small changes in the network environment render the pure execution Exec(JTP) (cyan dashed) ineffective, dropping nearly down to the performance of random plans at a SMAPE of 0.5, as presented in the absolute cost value results in Figure 26 top right. The execution suffers from complete inflexibility, not being able to react to changes. This deficit is visible from the RDS analysis of the execution in Figure 27 (e), showing a significant amount of unscheduled tokens, which leads to further violation of the minimum throughput requirements, compared to the cost-value-optimized plan. A huge imbalance of these two factors, owned by the attracting forces, to a significantly lower monetary cost, belonging to the repelling forces, shows that the execution Exec(JTP) fails in the decision whether to transmit data or not. Whenever a planned transmission cannot be completed as planned, the approach drops the data, reaching an additional data drop rate of 26.97% compared to an optimal plan, as visualized in Figure 26 (c). These results attest the deficits of the approach, leading to substantial inefficiency in transmission and evidence the need for a more flexible approach: adaptation.

Even for high network prediction errors, the adaptation approach is able to sustain more than one third of the performance gain from planning.

ADAPTATION. For the adaptation approach, the performance results are in the desired range between Joint Transmission Planning (JTP) using perfect prediction and Opportunistic Network Selection (ONS), converging towards the second. The NRS results of the two framing approaches sink slightly with rising network prediction error. We assume this to be caused from in imbalance in the network change model, which tends to increase the amount of available access network resources, decreasing the relative amount of data traffic, which is correlated with a slightly worse performance of the two approaches, as identified in Section 4.3.5.

Activating only the *extended data release mechanism* of the adaptation, which is designed to handle network changes, Ada(JTP)-rel (red dashed) sustains a benefit of 7.76% NRS for moderate prediction errors of 0.3, corresponding to 65.64% of the of the margin between ONS and JTP. Even for a high SMAPE of 0.5 a performance gain of 4.5% NRS, corresponding to 36.23% is achieved. The performance of the approach with *location reference* mechanism Ada(JTP)-rel-loc and the complete adaptation Ada(JTP) show equal performance to each other, slightly better than activating the first mechanism only. This improvement is achieved by the *location reference* mechanism, releasing all remaining data of flows in time slots in which the flows have been allocated in the plan, helping to fill unexpected additional network resources. As it can be seen from Figure 26 (c), the long-term delay rate of the adaptation sinks in comparison to the optimum and JTP, converging towards this of ONS. The Relative Detail Score in Figure 27 (c) confirms this behavior. Comparing the characteristics to those of JTP and ONS, the convergence



(a) Network prediction error

(b) Network prediction error

(c) Network prediction error

(d) Network prediction error

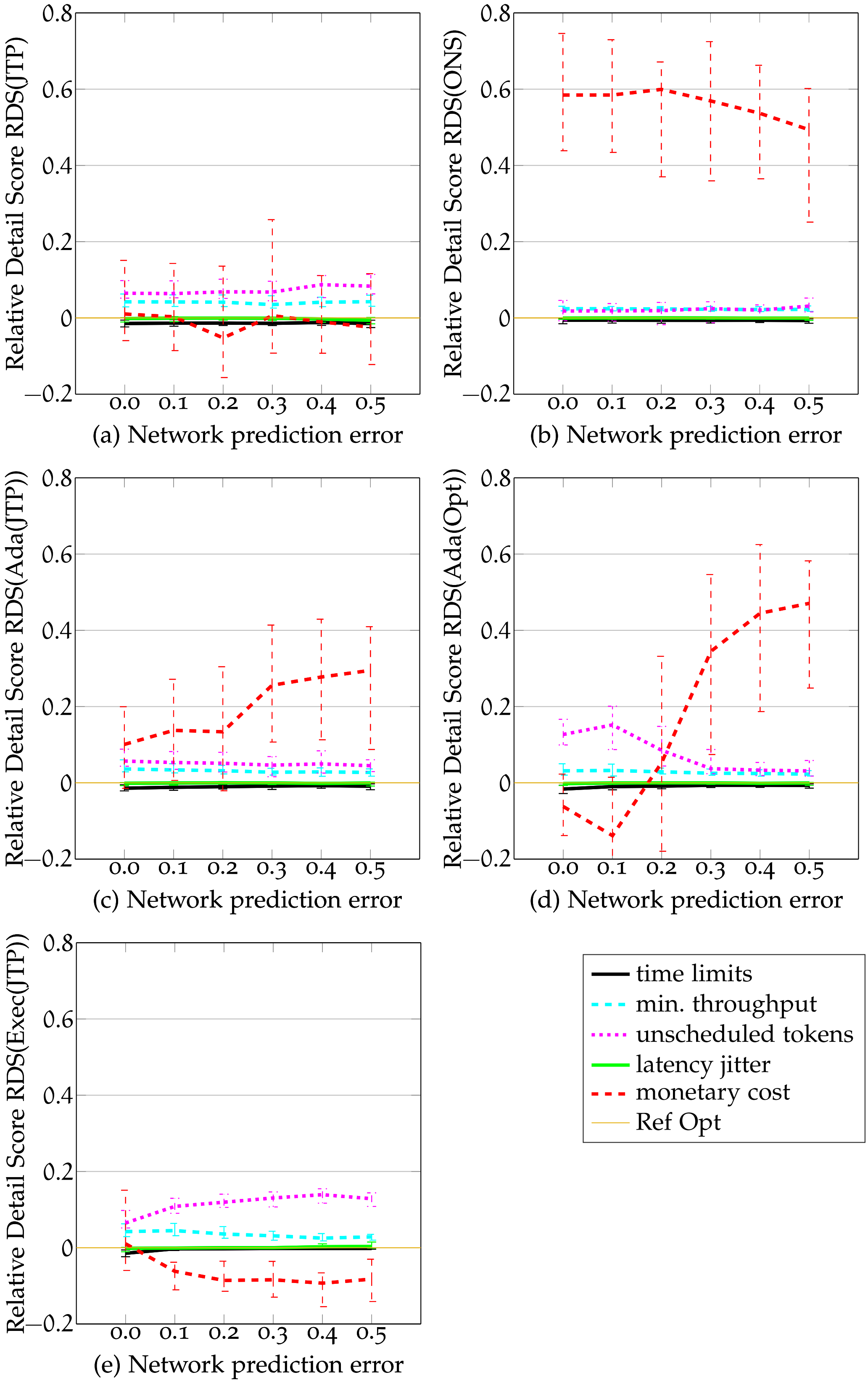
Figure 26: NRS, Cost function value, drop rate and execution time over network prediction error (SMAPE)

towards ONS gets obvious with slightly sinking values, except the substantially rising monetary cost criteria, indicating the loss of the superior temporal transmission pattern of JTP with stronger network changes, forfeiting purposeful transmission delaying. This effect claims that planned data for which transmission could not be completed should not always be released completely, as dictated from the mechanism, but rather be considered to be delayed further. However, this decision relies on an analysis of future available network resources, corresponding to planning rather than adaptation. Since the resulting performance loss is limited, only high network prediction errors justify triggering of a more complex re-planning using freshly predicted input data. For small and medium network prediction errors, the adaptation mechanism works effectively.

Optimal initial plan. Adaptation of an optimal plan Ada(Opt), according to Figure 26 (a), leads to worse results by 4-6% NRS in comparison to Ada(JTP) even though the initial plan is better. This is caused by an ambiguity of the optimum for long-term planning. For a planning horizon, the optimum for the first part alone differs from that of considering the complete plan. While the optimization considers all possibilities in the considered planning horizon to improve a long-term plan as a whole, ONS, the underlying approach of adaptation, optimizes the current time slot only with a statistical delaying. Optimization over time dimension creates long-term plans with a global optimum, referring to the entire planning horizon, compared to online approaches targeting local optima, referring to a single time slot only. When regarding a single time slot of a long-term plan, this disparity takes effect in apparent priority inversions, firstly, for data flows and, secondly, in the flow-network mappings. In the adaptation, the disparity of the approaches leads to a mix of the two, partially eliminating the global optimum, converging towards local optima. Since inversions are less often in JTP because of using the same network selection as ONS, it is easier for ONS to interpret the plan regarding efficient transmission. Hence, adapted plans from JTP outperform adapted optimal plans Ada(Opt) for network changes. Considering the execution time in Figure 26 bottom right, the disparity of approaches is also indicated from a higher computational effort of Ada(Opt) compared to Ada(JTP), reflecting more changes to the plan through adaptation. As follows, this can also be seen in the Relative Detail Score (RDS) analysis in Figure 27 (d). Heading to a characteristic similar to that of the execution mechanism at small prediction errors till a SMAPE of 0.1, the RDS of Ada(Opt) converges very fast towards the characteristics of ONS with low overall RDS values except causing a substantially higher monetary cost compared to the optimal transmission plan. This higher cost reflects the limited time selection capabilities of ONS, targeting local optima for single time slots. Despite these effects, the adaptation approach, when applied to plans from the optimal approach Ada(Opt), performs significantly better than ONS alone, profiting from the original temporal transmission patterns of the optimization.

Similarity of the selection algorithms for planning and adaptation lets an adapted plan from JTP outperform an adapted optimal plan in the presence of network changes.

Conclusions. The adaptation approach Ada(JTP) succeeds with handling network changes due to its *extended data release* mechanism, sustaining even for large SMAPE prediction errors of 0.5 a performance gain of 4.5% NRS (36.23%). For



(a) Network prediction error

(b) Network prediction error

(c) Network prediction error

(d) Network prediction error

(e) Network prediction error

Figure 27: Relative Detail Scores over network prediction error (SMAPE)

moderate errors of 0.3, it can even sustain a gain of 7.76% (65.64%). It profits from the similarity of algorithms for planning and adaptation, helping the approach to identify planned behavior as beneficial. Therefore, in the presence of network changes, the adaptation of plans from Joint Transmission Planning even leads to a better overall performance than the adaptation of an optimal plan.

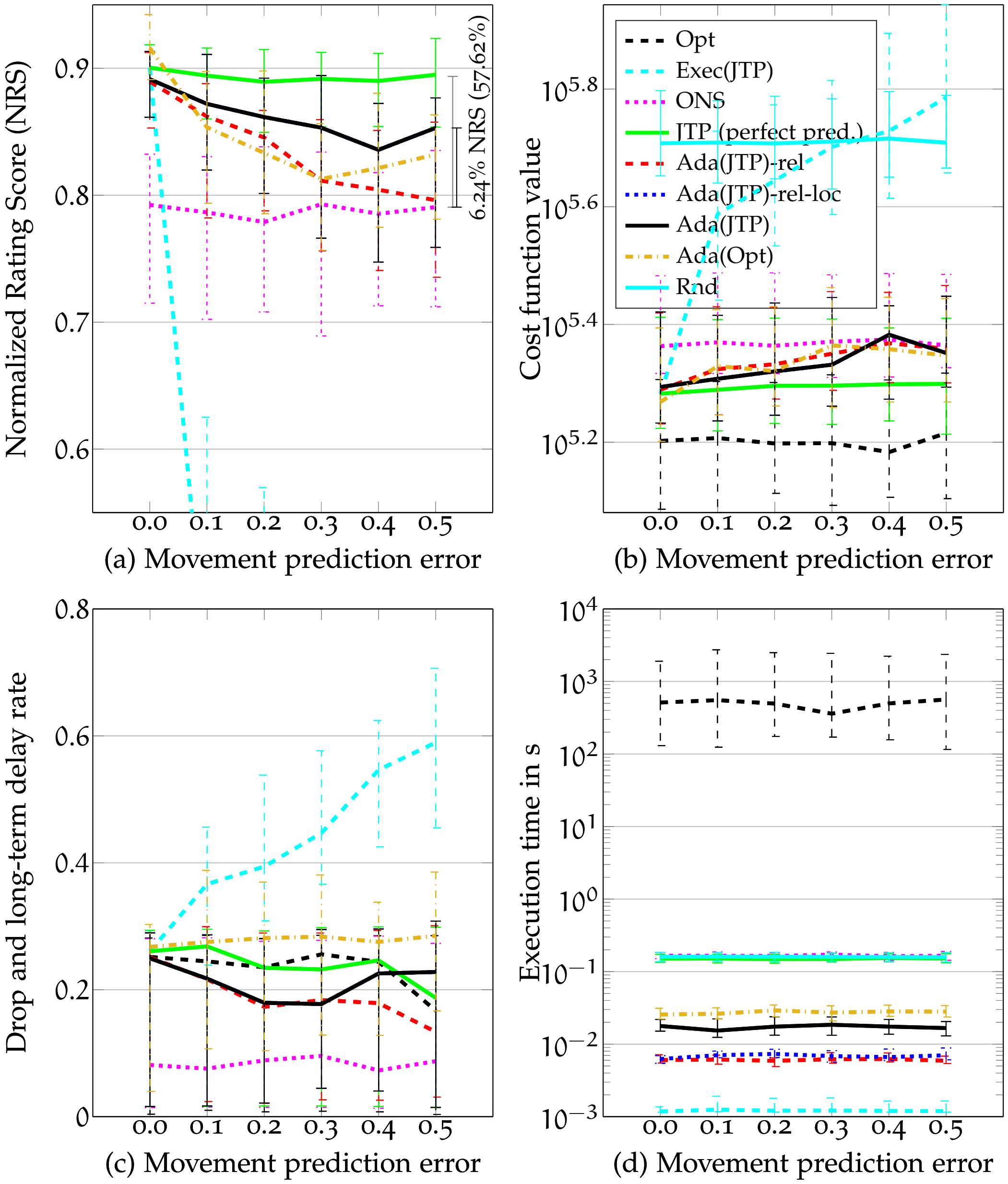
5.3.3 *Impact of Movement Prediction Errors*

The vehicle speed affects the time at which a node reaches a certain location, determining basic network availability and characteristics. If the node moves faster or slower than expected, the predicted network characteristics are shifted to another point in time, creating a temporal offset for experiencing the expected network environment. We analyze the effects of the movement prediction error on the adaptation and execution approach in the following.

execution. Pure plan execution shows, similar to the behavior in network prediction errors, a severely decreasing performance, even for small movement changes, as presented in the NRS results in Figure 28 (a). The absolute cost value of Exec(JTP) reveals that it performs even worse than the random approach in average for high movement prediction errors. The data drop rate presented in Figure 28 (c) and the RDS characteristic in Figure 29 (e) conforms to those of the network prediction error analysis, showing the execution's deficit of not being able to transmit data at a later point in time if transmission as planned failed.

adaptation. In contrast, the adaptation approach Ada(JTP) performs well, reaching NRS results framed by those of JTP with perfect prediction and ONS, as visualized in Figure 28 (a). 6.24% NRS over ONS, representing 57.26% of the margin between JTP and ONS for a large SMAPE movement prediction error of 0.5. These results attest the effectiveness of the *location reference* mechanism of the adaptation approach, targeting mitigation of the impacts of temporal offsets for experiencing network environments. In referring to the plan in a spatial dimension, the approach sustains the network selection from planning. To ensure compliance with temporal transmission requirements, like holding deadlines, the temporal offset for spatial reference to the plan is limited to the delay-tolerance of the specific data flow. Complying to that, the RDS results in Figure 29 (c) show close to optimal performance for the time limits criteria, while also keeping all other criteria low. Compared to the RDS characteristics of the JTP approach with perfect prediction, shown in Figure 29 (a), only the monetary cost characteristic shows significantly impaired results, still outperforming the corresponding results of ONS. The approach Ada(JTP)-rel-loc without the data flow prediction error handling performs equally. However, when only applying the extended data release mechanism, i.e. the Ada(JTP)-rel, the performance sinks significantly faster. Since networks are, due to the applied prediction error, in many cases not available for transmission as planned, the location the approach releases the data to be transmitted opportunistically, thus, converging towards the performance of ONS significantly faster.

The location reference mechanism sustains more than one half of the performance gain from JTP planning after adaptation in presence of strong movement prediction errors.



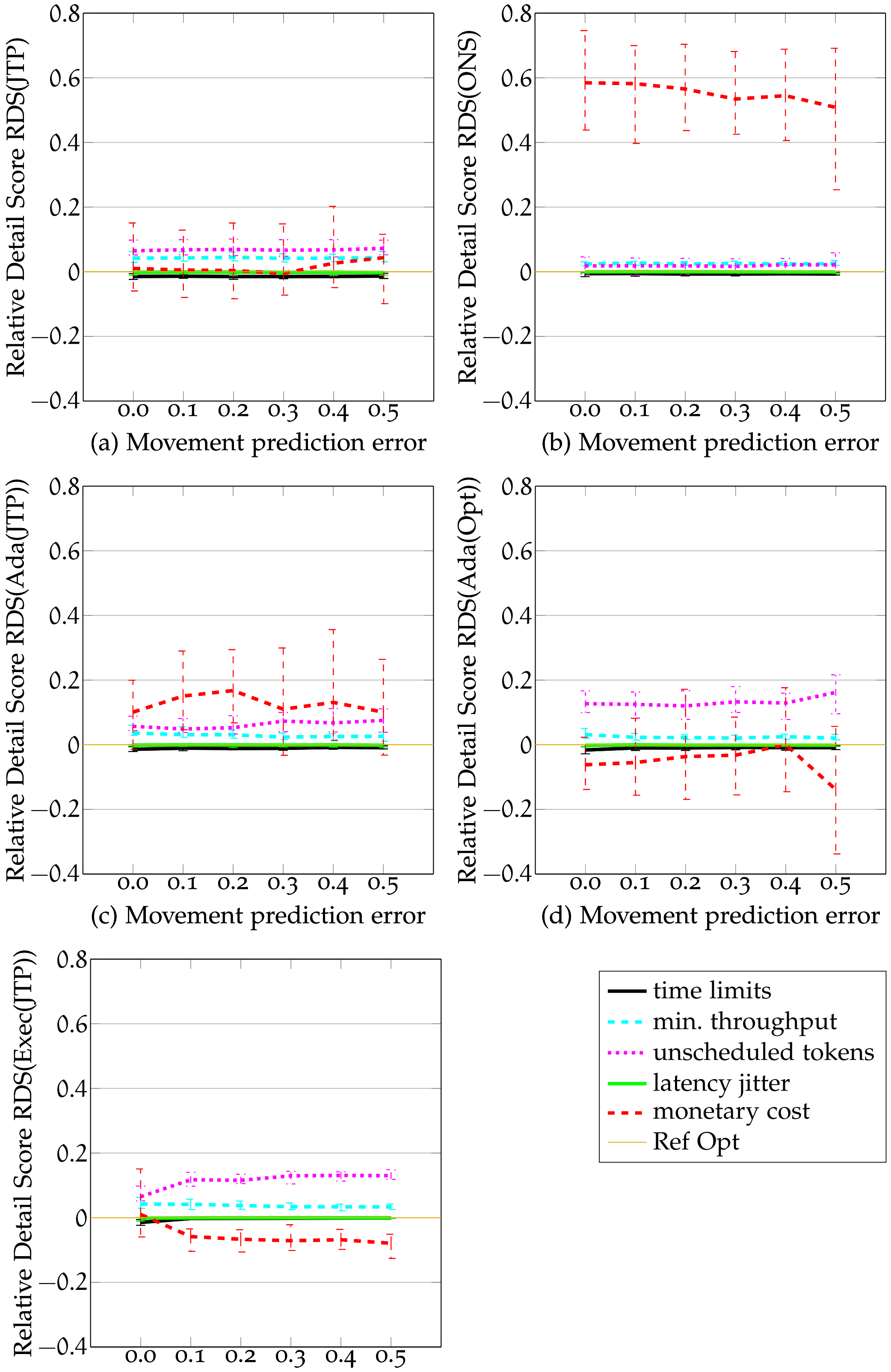
(a) Movement prediction error

(b) Movement prediction error

(c) Movement prediction error

(d) Movement prediction error

Figure 28: NRS, Cost function value, drop rate and execution time over movement prediction error (SMAPE)



(a) Movement prediction error

(b) Movement prediction error

(c) Movement prediction error

(d) Movement prediction error

(e) Movement prediction error

Figure 29: Relative Detail Scores over movement prediction error (SMAPE)

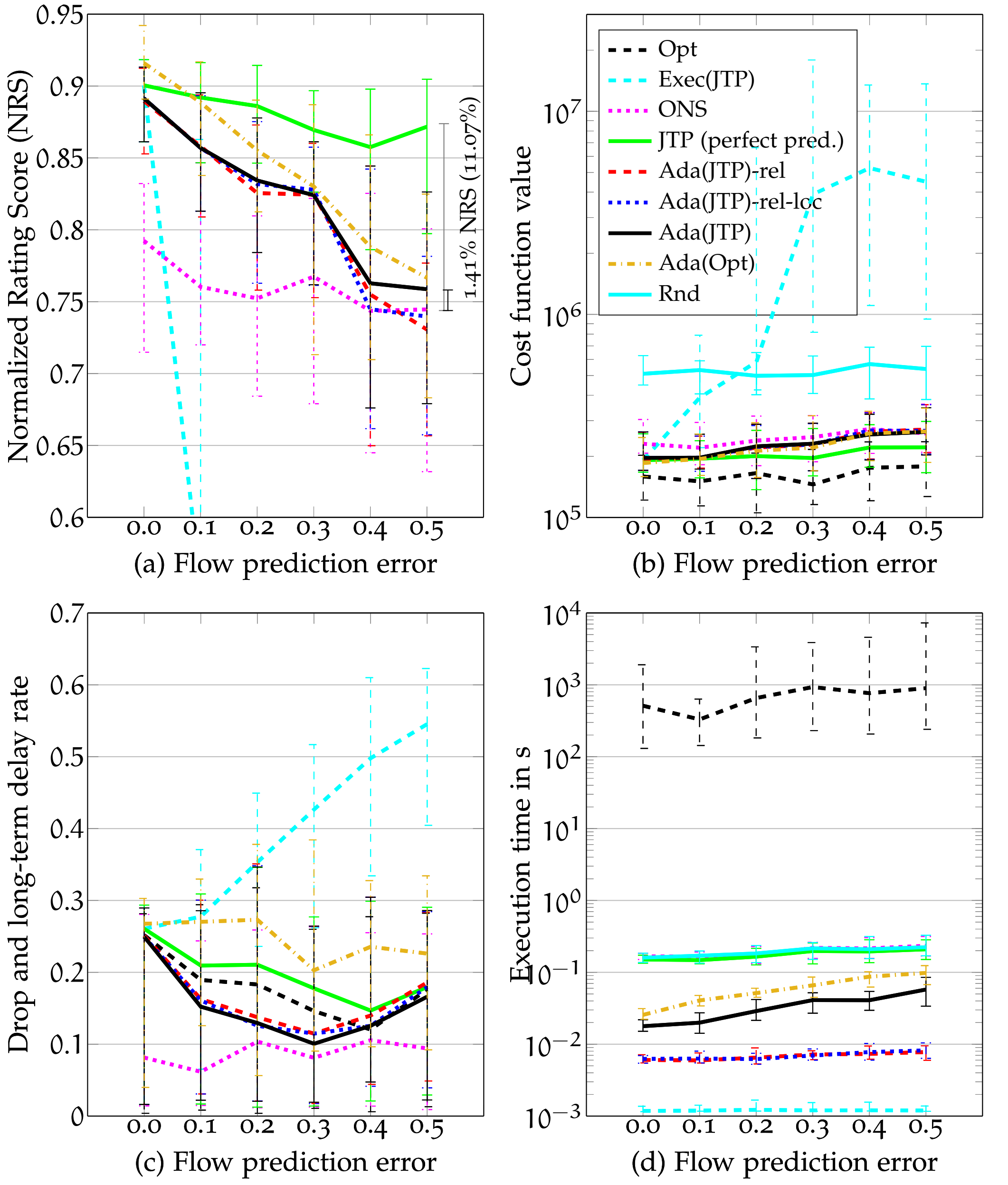
OPTIMAL INITIAL PLAN. Applying adaptation to an optimal plan Ada(Opt) does not reach the performance of applying it to the initially worse plan of JTP in the presence of movement prediction errors, as visualized in Figure 28 (a,b). Despite of the similarity of the performance characteristics with that of the network prediction error analysis, showing a rapid performance drop, the details behind this claim different reasons, as visible from the RDS analysis in Figure 29 (d). The RDS values for unscheduled tokens keeps constantly high, while the monetary cost keeps constantly low. In contrast to the effects in the presence of network prediction errors, the movement prediction error does not let the result converge against the local optima. The RDS characteristics stay constant and differ from those observed for ONS, as shown in Figure 29 (b), resembling the RDS results of the execution algorithm, which transmits less data than beneficial. This claim is confirmed by the data drop analysis in Figure 28 (c), showing an increased data drop rate for the Ada(Opt) approach compared to the optimal one Opt (black dashed). However, the difference is far lower compared that of the execution. Despite the data dropping effects, the approach Ada(Opt) is still able to sustain a gain of 4.14% NRS over ONS, corresponding to 41.90% of the margin between ONS and JTP.

CONCLUSIONS. The adaptation approach with its *location reference* mechanism is able to sustain 61.07% the performance gain from JTP over ONS for a moderate SMAPE movement prediction error of 0.3. It seems to go into saturation after that, sustaining still 57.26% at high movement prediction errors of 0.5. This outstanding result shows that movement prediction errors must be considered only as a minor factor in triggering a re-planning of the transmission. Furthermore, the analysis of adapting an optimal plan claims that the planning and adaptation approaches should go hand in hand in order to sustain the benefits reached in the plan efficiently. Accordingly, even though the initial plan of JTP is worse than optimal, the performance after adaptation in the presence of network and movement changes is significantly higher than that of an adapted optimal plan.

5.3.4 *Impact of Data Flow Prediction Errors*

User interaction or events triggering transmission in the background may add, cancel or delay data flows, resulting in differences in the planning. New data flows may interfere with planned traffic, competing for network resources, while canceled flows release additional resources usable for other data. In the following, we analyze the effects of data flow changes and how well adaptation mitigates their impact.

PLAN EXECUTION. For data flow prediction errors, we observe a strong impact on all approaches. The performance of execution Exec(JTP) shows a high variance and drops in median already for small SMAPE data flow prediction errors of 0.1 below the averaged random performance. It drops even further to about 10 times the cost of random approaches for moderate and high data flow prediction error strengths, as shown in 30 (a,b). An obvious reason for this is that new data flows are not covered in the plan and, hence, are not considered for transmission at all,



(a) Flow prediction error

(b) Flow prediction error

(c) Flow prediction error

(d) Flow prediction error

Figure 30: NRS, Cost function value, drop rate and execution time over data flow prediction error (SMAPE)

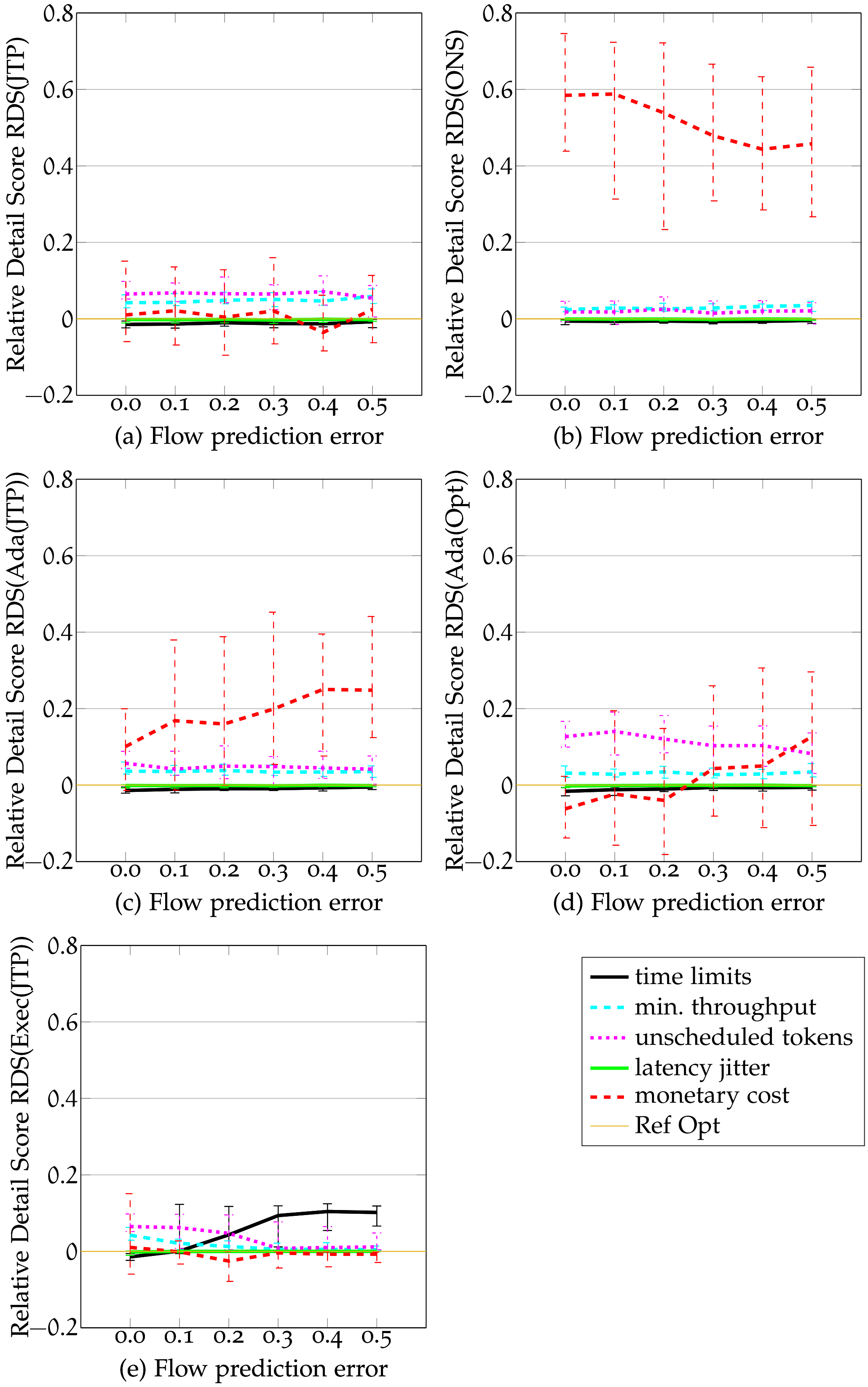
leading to a strong impact of their attracting forces. The data dropping analysis in Figure 31 (c) shows that 36.93% additional data is dropped from the execution Exec(JTP) with strong data flow changes, affecting the performance rating. However, the RDS analysis of Exec(JTP) in Figure 31 (e) reveals that this impact is dominated by another one. While all other RDS criteria converge towards zero with rising data flow prediction error, the time limits criterion reaches a stable value of about 0.1. Delaying data flows or stopping them earlier changes their start times and deadlines. However, if an execution follows the old deadlines, this leads to severe time limit violation, causing exceptional high repelling forces. Compared to this impact, the other criteria get negligibly small. This demonstrates the importance of re-evaluating data allocation of a plan before transmission, as performed by the adaptation approach.

As desired, the results of JTP with optimal prediction and ONS frame the performance results of adaptation for any data flow prediction error, illustrated in Figure 30 (a). However, this envelope shows a negative performance trend in NRS with an increasing variance during more data flow changes. We expect this to be caused by the trend of the prediction error model to add additional data flows, which complies with the negative correlation with the scenario optimization potential as identified in Chapter 4, shown in the Appendix A.5.4.

For new data that is not covered in the transmission plan, adaptation cannot sustain any planning gain. Hence, the transmission quality converges fast towards that of the underlying opportunistic approach ONS, providing a stable lower bound.

ADAPTATION. The characteristics of the adaptation for a rising data flow prediction error shows a strong convergence towards ONS, without reaching its lower results. At an error of 0.5, the sustained gain from planning Ada(JTP) shrinks down to 1.4% NRS, corresponding to 11.07% of the margin between ONS and JTP using perfect prediction. For a moderate data flow prediction error of 0.3, the gain is even 5.67% NRS (55.56%). In comparison to the sustained gains for the network and movement changes of 4.5% NRS (36,23%) and 6.24% NRS (57.62%), this is small. The adaptation's Relative Detail Score (RDS) characteristics consequently converge towards those of ONS with a higher monetary cost value and slowly falling other characteristics due to the relative decrease of the impact of these criteria. Regarding NRS results, the flow prediction error leads to differences between Ada(JTP) and Ada(JTP)-rel-loc, which does not integrate the *data flow prediction error handling* mechanism. The performance gain from this is negligible for small errors but mitigates the impact of strong errors above 0.3, sustaining a gain from planning and keeping performance above that of ONS, as visible in Figure 30 (a) However, the adaptation approach shows a slightly lower dropping rate for medium prediction errors, close to those of ONS, catching up again to the value of the optimum for higher errors. To give a visual impression of the strength of changes, a typical example scenario with SMAPE of 0.5 is shown in the Appendix A.6.

OPTIMAL INITIAL PLAN. In contrast to the other prediction error components, applying adaptation to an optimal plan Ada(Opt) in the presence of data flow prediction errors, sustaining a gain of 2.2% NRS (17.40%), outperforms the result of applying adaptation to a JTP plan Ada(JTP), as visible in Figure 30 (a). The data drop rate of Ada(Opt), presented in Figure 30 (c), is significantly higher than that



(a) Flow prediction error

(b) Flow prediction error

(c) Flow prediction error

(d) Flow prediction error

(e) Flow prediction error

Figure 31: Relative Detail Scores over data flow prediction error (SMAPE)

for the adaptation of JTP plans. This is also reflected in the RDS results, visualized in Figure 31 (d), showing a high but slowly sinking impact of unscheduled tokens on the result, accompanied with a growing impact of the monetary cost criterion. Besides that, constantly low RDS values confirm the similarity to ONS after applying adaptation targeting slot-wise local optima and increasing especially monetary cost. This characteristic indicates that with a higher prediction error, more data is allocated in a non-optimal way using ONS. Even though the trends in RDS and NRS are similar for adapting a plan from JTP and an optimal plan, the much higher drop rate of Ada(Opt) is prominent. We assume this to be caused by the different allocation strategies of ONS and the optimal one, leading to the disparity of global and local optima, as explained in the movement change analysis.

CONCLUSIONS. Data flow changes impair the performance at most of the three prediction errors. While, at a SMAPE flow prediction error of 0.3, the adaptation is able to sustain 5.67% NRS, corresponding to 55.56% of the margin between ONS and JTP with perfect prediction, the value sinks to 1.41% NRS (11.07%) at an error of 0.5. Thus, high flow prediction errors should be employed to trigger a re-planning. For adaptation of an optimal plan, the performance after handling data flow prediction errors is above that of adaptation of a transmission plan of JTP.

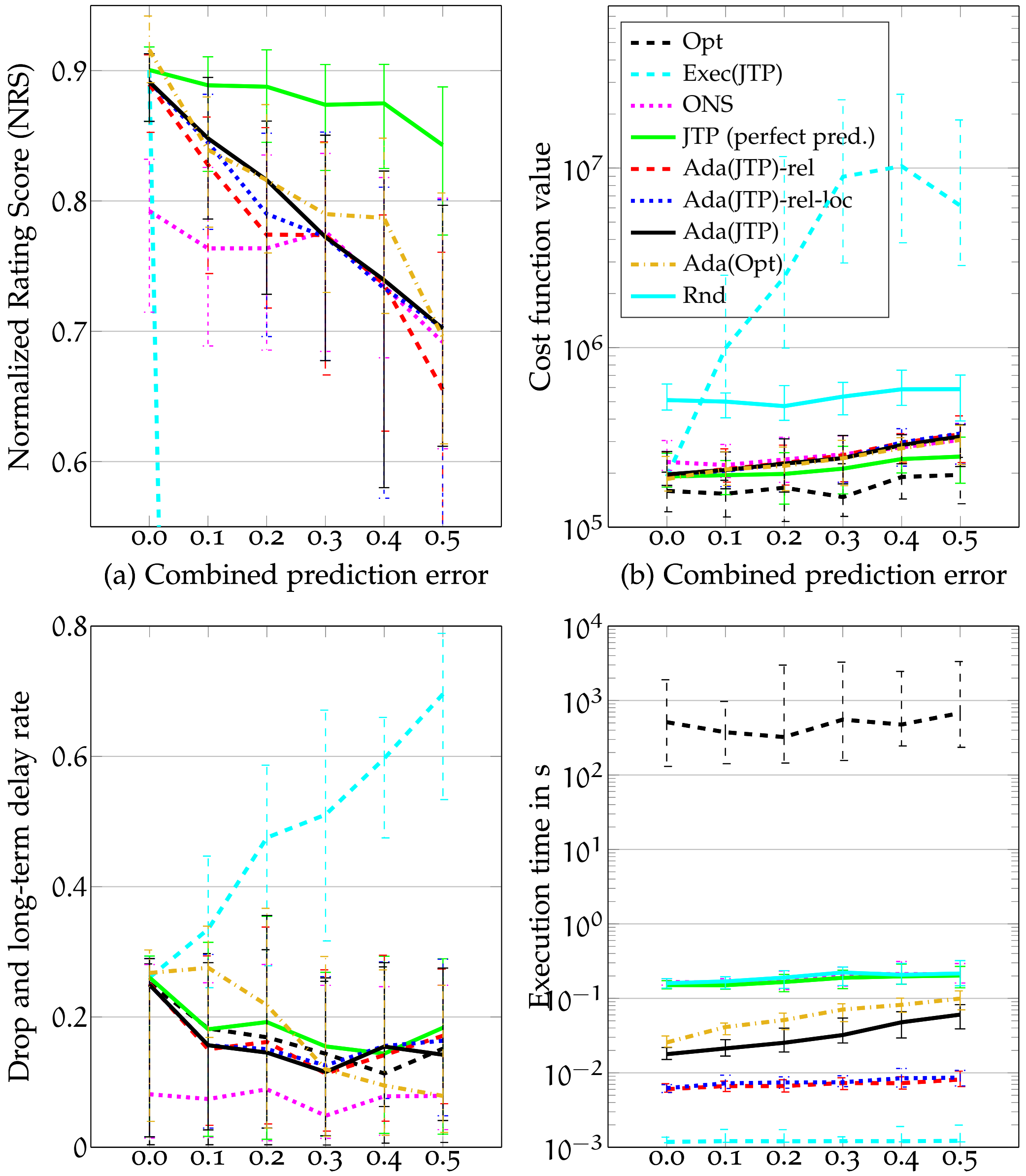
5.3.5 *Impact of Combined Prediction Errors*

In reality, environmental changes do not occur isolated from another. Hence, we create a combined prediction error, which simultaneously adds the three categories of environmental changes with equal strength to the scenario. In this section, we analyze the effect of this combination of errors and their impact on the adaptation approach.

The impact of flow changes dominates the performance reduction at combined prediction errors. The performance converges to that of ONS at a moderate error of 0.3.

The characteristics of the adaptation approaches for the combined prediction error are dominated by the data flow changes. The reason for this is the dependence of the changes on the plan. However, data flow changes add and remove data, reducing the share of transmissions that can be executed according to the initial plan. In contrast, network and node movement changes affect only planned data and reduce the efficiency of the transmission because, with increasing prediction errors, long-term planning decisions pursuing a global optimum are replaced by online decisions, approximating the local optimum for the current time slot for planning data. Accordingly, the two error components apply only to this reduced share of data, for which a reference in the initial plan still exists, thus, limiting their impact on performance and reasoning the impact dominance of data flow changes.

EXECUTION. For the execution Exec(JTP) and a SMAPE of 0.5, there are median data drop rates of 70%, as visualized in Figure 32 (c). Similarly to the effects of isolated data flow changes, this is dominated from time limit violations, as indicated from Figure 33 (e). A performance far below that of averaged random plans renders pure execution inefficient.



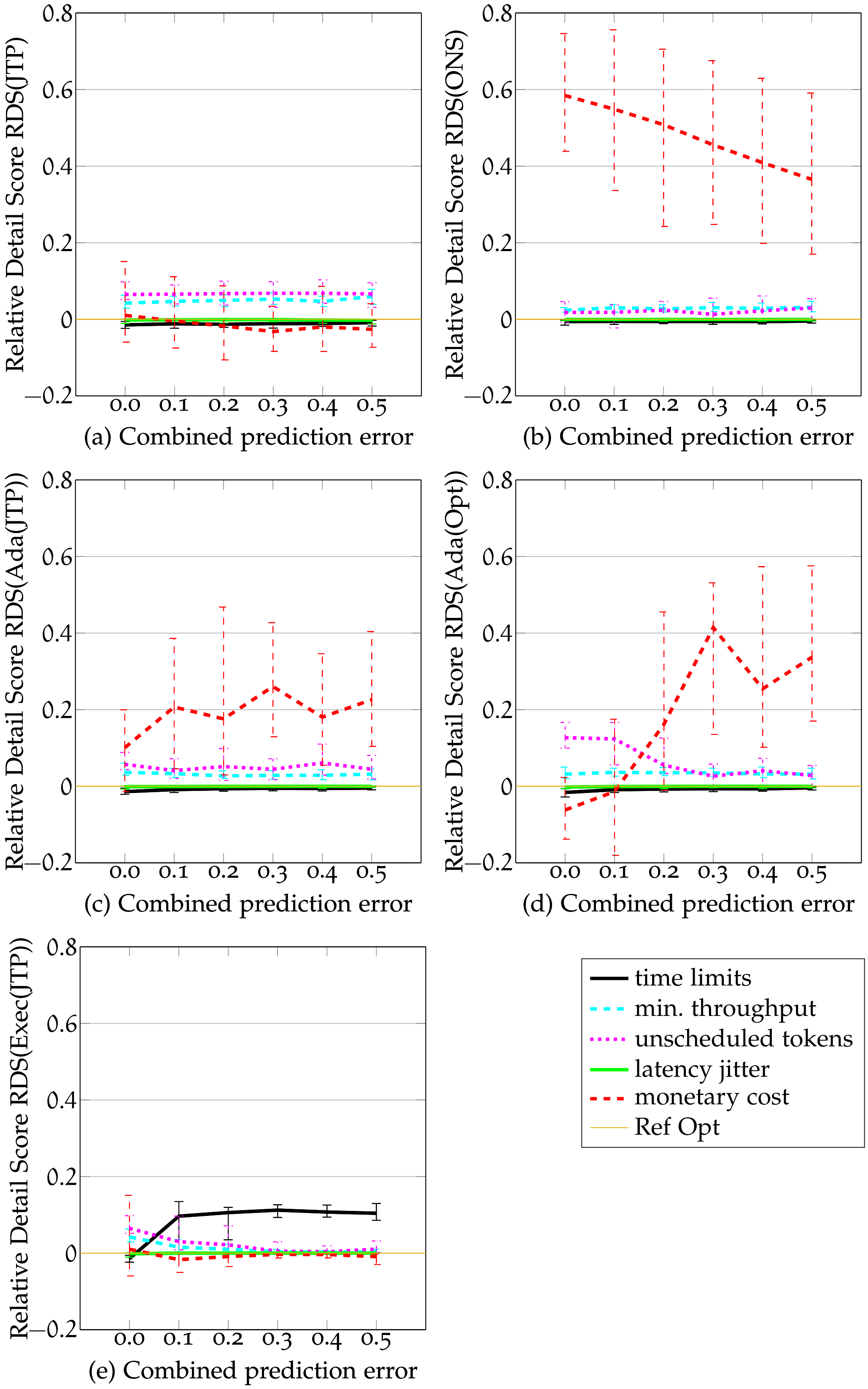
(a) Combined prediction error

(b) Combined prediction error

(c) Combined prediction error

(d) Combined prediction error

Figure 32: NRS, Cost function value, drop rate and execution time over combined prediction error (SMAPE)



(a) Combined prediction error

(b) Combined prediction error

(c) Combined prediction error

(d) Combined prediction error

(e) Combined prediction error

Figure 33: Relative Detail Scores over combined prediction error (SMAPE)

adaptation. As for the precedent change analyses, the performance results of JTP with perfect prediction and ONS construct a frame around the adaptation approach results, as depicted in Figure 32 (a,b). They follow a decreasing trend as seen for the isolated data flow changes. With increasing data flow changes, the performance of the adaptation approach Ada(JTP) sinks in median about linear in NRS, however with increasing variance. It converges towards the resulting quality of ONS, joining it at a SMAPE of 0.3, after which there is no significant gain from planning in comparison to applying ONS alone. Not falling below the performance of ONS, the adaptation demonstrates that it passes at least as much data as required to be treated from ONS. Even though the overall performance joins that of ONS, the resulting transmission is not identical. As shown in Figure 32 (c), the data drop rate of Ada(JTP) differs significantly from the data drop rate of ONS, sinking from being close to the optimal to the center between optimal and ONS. The Relative Detail Score (RDS) results, presented in Figure 33 (c), resemble a mix of the characteristics of network and data flow prediction errors with slowly rising monetary cost RDS values but high variances. Other RDS values constantly stay low, meaning that, compared to an optimal plan, monetary cost violations dominate the performance impact, followed by unscheduled tokens and minimum throughput requirement violations. This result shows that a SMAPE of 0.3 for each prediction error qualifies re-planning.

initial optimal plan. In comparison, the adaptation of an optimal plan Ada(Opt) shows similar performance with a positive peak at SMAPE 0.4 and, more important, a lower variance of the results. Hence, even though the performance of Ada(Opt) is worse in isolated assessment after applying environment changes for the network environment and node movement, in the combined error analysis, it outperforms adaptation of a JTP plan. The mechanisms of adaptation handling network and movement changes can only sustain a gain from data, for which a reference exists in the initial plan. Hence, after applying data flow changes, these two lose their effect for changed data flows, explaining impact dominance of data flow changes over the other two. Furthermore, while the data drop rate for adaptation of JTP follows that of its initial plan, adapting the optimal plan converges towards ONS with an initial tendency to the pure execution. As follows, this is confirmed by the RDS analysis in Figure 33 (d). While at low SMAPE errors being dominated from a high data drop rate, i.e. the RDS value unscheduled tokens, the characteristics converge towards the RDS results of ONS showing a high monetary cost value. As illustrated in Figure 32 (d), the additional efforts for adapting the initial plans are reflected by a rising execution time, indicating the higher change rate through adaptation.

For the combined error, adaptation of an optimal plan slightly outperforms adaptation of a plan from JTP.

discussion. The combined error analysis demonstrates the feasibility of the adaptation approach. For simultaneously occurring moderate prediction errors, the adaptation is able to sustain a gain from initial transmission plans, while even after convergence to ONS, the approach does not perform worse. With combined environmental changes, the data flow prediction error dominates the result, reducing the impact of the other two, since they affect only data transmissions, which comply to the plan. Hence, the data flow prediction error is the most important factor for considering re-planning.

5.3.6 *Execution Time Analysis and Re-Planning*

The measured execution times provided in the previous sections cover the execution of all time slots of the planning horizon. However, online algorithms, like ONS and the adaptation are executed consecutively slot by slot, each at the required point in time, using the current state environmental state. Hence, we normalize the execution time of these online approaches by the number of time slots in Figure 34 for the scenario of 8 networks and 8 data flows. Accordingly, after an initialization overhead, the planning approach JTP follows a linearly rising characteristic increasing its execution time in average by a factor of 2.54 for doubling the number of time slots. At a planning horizon of 1600 time slots, it reaches a median execution time of 227 ms. For a re-planning, the time required to refresh prediction, maybe based on external data sources, has to be considered as well. We expect it to be achievable below one second, or two seconds if accessing external databases. Hence, a reaction to severe environmental changes can be achieved very fast. However, for interactive or multimedia applications, this additional delay is by far too long [200]. For these cases, adaptation fills the gap. Reaching, after a first initialization, an execution time for one time slot of less than 250 µs, see Figure 34, even in this non-optimized implementation, the approach is responsive enough to handle sudden environmental changes instantly. The SMAPE prediction errors quantify the strength of these changes and may be employed to implement dedicated trigger conditions for re-planning. Since the performance of the adaptation approaches stayed above the performance of ONS, our combined planning and adaptation approach defines an improvement to transmission performance of mobile nodes using heterogeneous networks.

Adaptation enables sub-millisecond decisions to cure inefficiencies in the planning in the presence of prediction errors. Since data flow changes dominate the performance impact, they should be employed as the main trigger for re-planning.

5.4 SUMMARY AND CONCLUSIONS

We designed and evaluated a data transmission plan *adaptation* approach Ada(p), which effectively mitigates the impact of environmental changes on transmission performance, enabling the application of transmission plans. The adaptation employs an underlying opportunistic transmission approach. It constraints the opportunistic decision variable whether to transmit or not in order to follow an initial transmission plan, called plan execution Exec(p). In order to react to environmental changes, we designed three mechanisms that each handles one kind of prediction error and selectively release or modify constraints to allow the underlying opportunistic transmission approach to adapt the plan.

The mechanism treating network changes releases data for opportunistic transmission, which was either planned for allocation now or at an earlier time slot or not at all. Thus, the mechanism can handle network capacity fluctuations and fill unused resources with unplanned data. The mechanism treating movement errors extends the first by modifying the method to refer to the plan. Instead of referring to the current time slot of the plan (planning happens in the time dimension), it

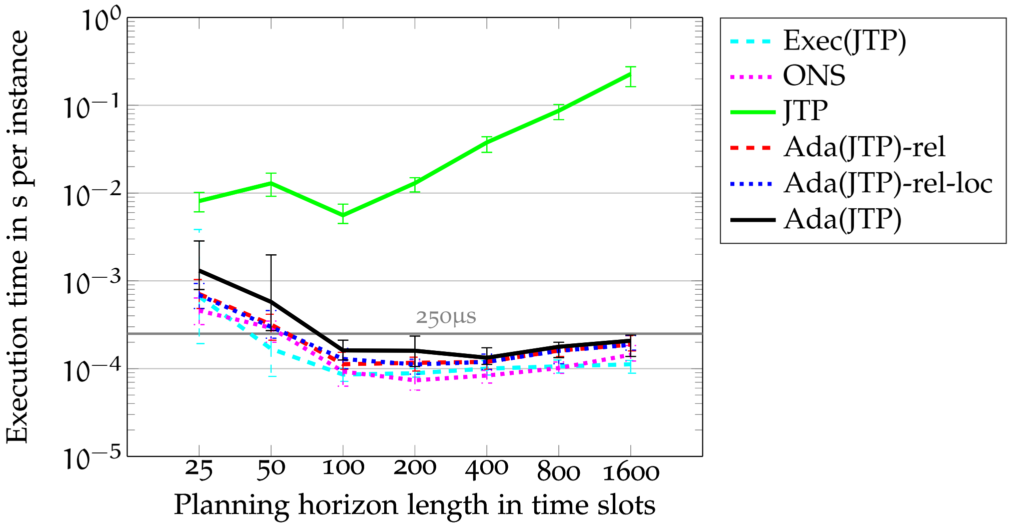


Figure 34: Execution duration for one instance of the approach, corresponding to the complete time horizon for planning and one time slot for online approaches

refers to the time slot in the plan that corresponds to the current *location* of the vehicle. For delay-tolerant transmissions for which transmission time selection is less important, this preserves the network selection pattern of the initial plan. This location reference causes a temporal offset of the planned transmission. To sustain temporal transmission requirements of data flows, we limit this offset individually for each data flow by its delay-tolerance. The third mechanism treats data flow changes. For these kinds of changes, there exists no reference in the plan to sustain. Hence, we transmit unplanned data flows opportunistically and, for rising errors in the current time slot, gradually decrease the threshold of the opportunistic transmission to follow the initial plan, converging towards complete opportunistic transmission at high data flow changes.

We apply this approach to plans of Joint Transmission Planning (JTP) and optimal plans and use Opportunistic Network Selection (ONS) as the underlying opportunistic approach. The adaptation is able to sustain a significant share of the performance gain from planning over ONS. For evaluation, we employed isolated prediction errors to create environmental changes for the networks, the movement and the data flows, varying them in strength between 0.0 and 0.5 (Symmetrical Mean Average Prediction Error). Finally, we combined them due to applying all three kinds of errors with equal strength to the same scenario. We present the four main conclusions from this evaluation, ordered by their importance.

- For moderate isolated prediction errors of 0.3, the adaptation Ada(JTP) sustains more than a half of the gain from JTP over ONS, in particular, 65.64% for network, 61.07% for movement and 55.56% for data flow prediction errors. Compared to that, the performance of an execution Exec(JTP) without adaptation sinks far below the performance of ONS already for prediction errors of 0.1. These results demonstrate that the adaptation is able to cope with prediction errors effectively.

- For strong, isolated prediction errors of 0.5, the Ada(JTP) still sustains 57.62% for movement, 36.23% for network and 11.07% for data flow prediction errors over ONS. The convergence towards the performance of ONS is especially strong for data flow changes because there exists no theoretical gain from planning. In particular, data flow changes dominate the performance when applying the combined prediction error, converging to the performance of ONS at an error strength of 0.3. In contrast to Exec(JTP), the adaptation Ada(JTP) does not sink below the performance of ONS, providing a stable lower bound performance.

- The adaptation is applied to single time slots. After initialization, it reached a per-time-slot execution time of less than 250µs in our tests, which enables a responsive reaction to environmental changes without impairing latency requirements of applications. In contrast, the planning approach JTP always has to plan the entire time horizon, reaching a median execution time of 227.34ms for a planning horizon of 1600 time slots and is too inefficient for instant reactions to environmental changes. This proves the necessity of adaptation with a processing time, which is lower by three orders of magnitude, and attests its feasibility for real systems.

- Applying adaptation to an optimal plan Ada(Opt) leads in many cases to worse results than Ada(JTP). We assume this phenomenon to be based on the similarity of JTP and ONS, applying the same search ordering heuristics for data flow and network prioritization. Thus, when data is released to be allocated opportunistically using ONS, the probability is higher for Ada(JTP) than for Ada(Opt) that the adaptation comes to the same or a similar result as in the plan. For Ada(Opt), it applies more changes and may create inefficiencies due to destroying optimal transmission patterns and building less compatible opportunistic ones. Thus, for the development of new transmission planning strategies, we recommend analyzing the performance after adaptation as well. In the case of detected inefficiencies, a better compatible opportunistic approach as an underlying algorithm for adaptation should be derived from the planning strategy.

Our adaptation demonstrates that it can effectively sustain a significant share of the performance gain from planning responsively. In the next chapter, we present our novel mobility management protocol MoVeNet, which pools available wireless network resources for a mobile node and enables dynamic data flow distribution via the networks. It represents a platform for execution of transmission plans, as presented in this and the previous chapter.

6 MOBILITY MANAGEMENT FOR TRANSMISSION PLAN EXECUTION

This chapter addresses user-centric and flexible utilization of cross-operator network resources, achieved with the design and development of our novel protocol Mobility Management for Vehicular Networking (MoVeNet). The previous chapters focused on strategic transmission planning and their adaptation to optimize the data transmission for vehicles. However, the conventional Internet network stack does not support the execution of those transmission plans because switching from one access point to another, i.e. performing a handover, breaks a running connection. The reason for this is the double role of IP, as detailed in Chapter 3.3. An assigned IP address affiliates the mobile node to an access network, enabling packet routing. In addition, the communication partner uses the IP address for identification of the mobile node. Hence, a change of the IP address during handover, as desired, updates the affiliation of connections to the access networks but, as an unwanted side effect, invalidates the identification of any open connection. All open connections break and have to be reinitialized to continue or restart the transmission.

MoVeNet *makes cross-operator network resources available for flexible use and, thus, enables application of advanced transmission strategies.*

To target this problem, researchers have developed various so-called mobility management approaches. Famous ones are Mobile IPv6 [164] and Multipath TCP (MPTCP) [73]. The basic concept, which all of them have in common is introducing a new persistent identifier for the client or connection to which actual IP addresses are mapped dynamically in order to achieve the desired packet routing. Hence, the identifier stays valid even though the IP address changes and the connection can continue its data transmission. Screening the landscape of existing protocols in Section 3.3 reveals a gap for a protocol, which satisfies functional as well as non-functional requirements imposed from strategic transmission planning and the connected vehicle scenario. Protocols, e.g., do not support cross-operator handover and multi-homing sufficiently (PMIPv6 [83]), do not cover the entire IP data traffic (MPTCP) or lack compatibility with legacy nodes (HIPv2 [151], SHIM6 [159]). To analyze the gap further, we collect functional and non-functional requirements for enabling efficient transmission plan execution in the connected vehicle scenario in the next section. Based on these requirements, we compose a novel distributed communication architecture together with a protocol design, introducing new concepts for IP mapping, packet re-sending and complementary signaling mechanisms.

The chapter is based on our publications *MoVeNet: Mobility management for vehicular networking* [182], *Publish-subscribe-based control mechanism for scheduling integration in Mobile IPv6* [185] and *A Concept for Vehicle Internet Connectivity for Non-Safety Applications* [184] as well as the supervised student theses and works [105, 6, 19, 215, 134]. The chapter revises and extends the therein presented concepts and analyses. We evaluate the approach analytically, in simulation and with a prototype, proving protocol feasibility and present its performance and new features. Finally, we give considerations for integration of transmission planning, as presented in Chapters 4 and 5, into the distributed architecture of MoVeNet.

6.1 SYSTEM REQUIREMENTS

To realize data transmission planning, the mobility management approach has to meet multiple functional requirements:

F-Req-1 **All IP traffic.** Data traffic management treats incoming as well as outgoing packets, especially of TCP and UDP data flows as well as ICMP packets. All non-managed traffic affects the available network resources and distorts the data traffic management's result. Accordingly, the approach should cover the entire IP data traffic of the client. Transport layer approaches, like Multipath TCP [73] or mSCTP [117, 114], cannot satisfy this requirement.

F-Req-2 **Routing flexibility.** Distribution of data flows over multiple networks boost transmission performance, as shown in Chapter 4.3.4. To route data concurrently via multiple interfaces, multi-homing techniques have to be applied. In addition to traffic distribution, the handover of individual data flows, rather than treating all traffic as a unity, is an important feature that enables spreading of data flows across parallel networks. Whenever a data transmission plan requests it, the mobility management approach has to move a specific data flow from one network to another.

This requirement is not satisfied from network-controlled approaches, like Proxy-MIPv6 [128, 78] or LISP [71] as well as most IP-based protocols, e.g. most MIPv6 derivatives, which do not distinguish between different transport layer connections but handle all data traffic as a unity.

F-Req-3 **Compatibility.** As stated in *F-Req-1*, the approach shall grant access to all media on the Internet. Since most servers in the Internet are not assumed to implement new features timely or at all, mobility management approaches should be able to communicate with conventional network stacks. A lack of this kind of compatibility is given for direct mobility management approaches, like MPTCP [73], SHIM6 [170, 169] or HIPv2 [151], as each communication partner has to implement the required network stack modifications.

In addition, the fast-changing perceived network environment and desired applications in the connected vehicle scenario impose tough non-functional requirements. Firstly, these are safety-supporting services, which require low latency [65]. Secondly, it covers multimedia applications, conversational data traffic or mobile office applications, which require high throughput, low latency or continuous data transfer [200].

NF-Req-1 **Low-latency routing.** The connected vehicle scenario covers safety supporting applications and conversational multimedia traffic, profiting from low latency access to the Internet. Hence, the architecture is supposed to provide low-latency routing to the corresponding communication partners. For critical data flows, routing structures that introduce additional routing delays to the end-points should be avoided.

NF-Req-2 **Low-latency handover.** The connected vehicle scenario is characterized by high node mobility. Especially when using short-range technology, like WiFi, connection duration is supposed to be short, as detailed in Chapter 2.2.4. In particular, handovers of critical data flows between networks have to happen fast and seamlessly to allows controlling mechanisms to react rapidly to unexpected environmental changes.

NF-Req-3 **Low signaling overhead.** The object under optimization is the Internet access. Additional signaling overhead drains the available access network resources and, thus, contradicts to the goal. Hence, the signaling overhead of the protocol should be as low as possible.

NF-Req-4 **High scalability and robustness.** The number of vehicles, which are connected to the Internet, is expected to rise to 90% till 2025 [62, 92], leading to an explosion of required peek network resources [58]. To be able to cope with this amount of connected vehicles, the approach must be scalable. In addition, the robustness of the approach must be ensured to mitigate the effect of faults or attacks on the system.

The vehicle scenario imposes extreme requirements to mobility management. High vehicle speeds and usage of short-range access networks lead to short network connectivity duration and, therefore, frequent handovers. In addition, many vehicle applications profit from a low latency connection [65]. Besides that, vehicles as well as other mobile users both profit from the satisfaction of the presented requirements. As presented in Section 3.3, existing protocols lack essential functions for realizing execution of data transmission plans. To overcome those shortcomings of other protocols and satisfy the presented requirements, we present the protocol *Mobility Management for Vehicular Networks* (MoVeNet), which is inspired by different mechanisms of existing protocols that have proven to be efficient and introduces new ideas for complementary mechanisms enabling flexible and distributed cross-operator mobility management.

6.2 MOVENET ARCHITECTURE

MoVeNet is a user-centric distributed IP mobility management protocol on the network layer, incorporating complementary mechanisms for IP table synchronization and connection initialization to optimize either for overhead or latency, depending on the connection requirements. It provides flexible data distribution capabilities to different routes letting dedicated scheduling algorithms, like the Transmission Planning and Adaptation presented in Chapters 4 and 5, control network selection.

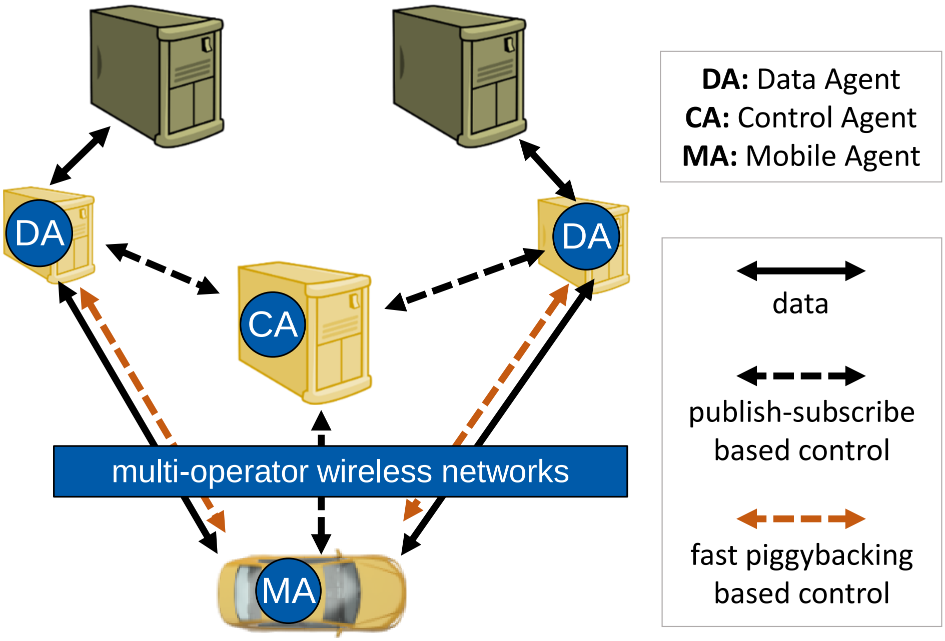

DA: Data Agent
CA: Control Agent
MA: Mobile Agent
DA
DA
CA
MA
multi-operator wireless networks
data
publish-subscribe based control
fast piggybacking based control

Figure 35: Architecture of MoVeNet. Data Agents act as proxies, hiding the protocol from correspondent nodes. Management functions are taken over by the Control Agent.

It consists of three kinds of entities, as illustrated in Figure 35: the Mobile Agent, Data Agents and a Control Agent. For connections between a mobile node and a correspondent node, the Mobile Agent in the vehicle establishes handover-enabled data bridges to one or more Data Agents located in the backend, which serve as proxy nodes, hiding the protocol from external nodes. The Control Agent takes over responsibilities for the distributed system management for the Mobile Agent, reducing signaling overhead via the wireless links. In the following, we first present MoVeNet's distributed architecture and explain the responsibilities of its entities.

6.2.1 *Mobile Agent – The In-Vehicle Entity*

The Mobile Agent is located at the mobile node and provides a virtual access network, capturing IP packets to enable flexible routing via desired routes and hiding mobility management from higher communication layers. It inserts and removes MoVeNet headers and manages the connections to the Control and Data Agents.

6.2.2 *Data Agent – Light-Weight Proxy*

A Data Agent is a backend entity for data routing, acting as a proxy and end-point for MoVeNet, which converts data packets from the Mobile Agent into conventional ones and conventional packets from correspondent nodes into MoVeNet

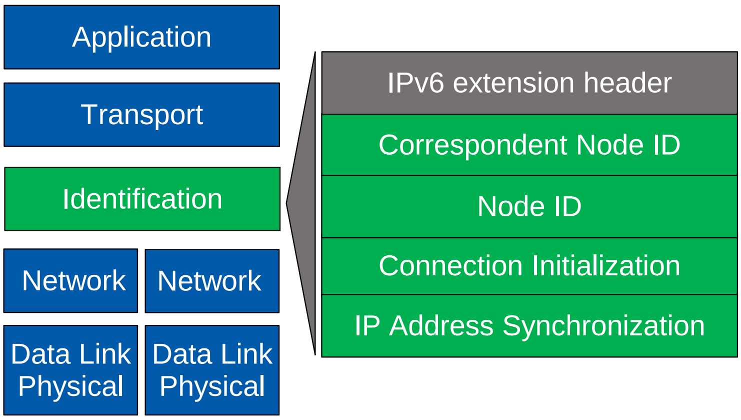


Figure 36: Simplified MoVeNet Identification layer with all optional fields

packets, hiding the protocol from external nodes. Hence, it provides substantial compatibility, accounting for requirement *F-Req-3*.

A Mobile Agent uses multiple Data Agents at the same time to optimize for specific connection requirements of each data flow. This corresponds to the concept of distributed mobility management, as presented in [226, 76] and detailed in Section 3.3. As a result, Data Agents can be selected with respect to their location, optimizing closeness to the optimal routes for individual connections and, thus, ensure low-latency for routing according to *NF-Req-2*. Furthermore, selecting a Data Agent close to the Mobile Agent reduces handover latency complying to *NF-Req-1*. Data Agents serve as anchor points for multiple vehicles and form a distributed network, which provides high scalability and robustness through redundancy, addressing requirement *NF-Req-4*.

Data Agents act as proxies, advisedly located to optimize route characteristics and to provide compatibility to conventional network stacks, while the Mobile Agent takes over management tasks for the Mobile Agent.

6.2.3 *Control Agent – System Orchestration*

The Control Agent complements the distributed architecture of MoVeNet with an entity, which takes over essential management functions from the Mobile Agent, such as Data Agent selection, initialization and maintenance as well as IP address synchronization. Thus, the Control Agent orchestrates a network of Data Agents for the Mobile Agent, forming a partially distributed mobility management architecture for MoVeNet.

6.2.4 *Routing and Signaling Concept*

MoVeNet employs complementary management mechanisms, which are applied depending on the required management characteristics of each data flow, in particular regarding latency. For flexible data routing between the Mobile Agent and Data Agents, two functions are required. The first function is the announcement of available routes, i.e. wireless access networks, for packet dispatching to Data Agents. For this purpose, MoVeNet introduces two complementary mechanisms. MoVeNet's default mechanism sends IP addresses, identifying new available routes, to the Control Agent, which serves as a publish-subscribe broker. It distributes this information to each active Data Agent of the Mobile Node, as illustrated in Figure 35 with black dashed arrows. The new routes can be used, as soon as synchronization to all active Data Agents is finished and acknowledged from the Control Agent to the Mobile Agent. Compared to other approaches, sending along signaling information for each connection [73, 169], this approach reduces the overhead via the wireless link to a minimum, in trade-off for a slightly longer delay for the announcements. The complementary route announcement mechanism of MoVeNet accounts for low-latency, carrying signaling information piggybacked to data packets and, thus, sending it on a direct route to Data Agents. It is designed for connections that require fast handovers.

Complementary route announcement mechanisms provide either a low overhead or, for selected connections, a low latency.

The second function addresses flexible packet transmission along announced data routes for individual data flows, establishing the above mentioned handover-enabled data bridges between the Mobile Agent and Data Agents. Therefore, MoVeNet employs a newly designed Identification header, inspired from HIPv2 [151], which is located between network and transport layer, as shown in Figure 36. MoVeNet modifies the packet's IP addresses to route data via the desired network to the Data Agent. To be able to recover the original connection context and continue routing to the target nodes, the Identification layer contains two identifiers in the optional fields Correspondent Node ID (CN ID) and mobile node ID (Node ID). Furthermore, there exist two additional optional header parts for connection initialization and IP address synchronization. We detail the exact functions of these headers and their conditions of presence in the following sections.

6.3 MOVENET PROTOCOL DETAILS

In this section, we first discuss MoVeNet's IP address mapping system between the Mobile Agent and the Data Agents, implementing the handover-enabled data bridge. We detail, which information is required at the corresponding entities to enable handover operation. Consecutively, we explain the Agent and connection initialization as well as the IP address table synchronization mechanisms, which transfer the required information to the entities. Finally, this section covers further enhancements of and considerations for MoVeNet, completing the protocol specification.

6.3.1 *IP Address Mapping System*

Mobility management is based on providing stable end-points to the client and corresponding node to hide changes of the network access from upper layers. To provide hidden routing capabilities, the Mobile Agent and the Data Agent both provide permanent IP addresses, establishing a handover-enabled data bridge to which dynamically changeable addresses for routing can be mapped as desired. For mapping, the packet must be identified and modified at four stages for one round trip according to Figure 37. For each packet transfer direction, this corresponds to one modification to enable routing and a second modification to undo, i.e. hide, the first modification from the receiver.

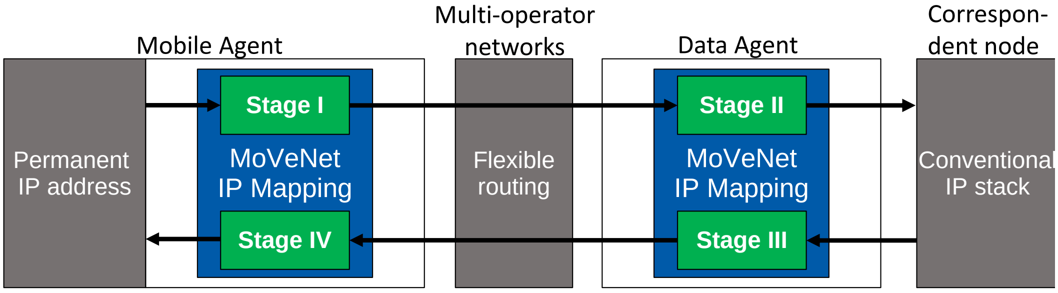


Figure 37: Packet processing stages of MoVeNet to provide hidden routing capabilities

In the following, we explain tunneling, representing the default mapping approach of proxy-based mobility management protocols, e.g. applied in MIPv6, and identify its shortcomings. Based on this, we developed a new mapping approach, which configures multiple IPv6 addresses on the same network interface to multiplex data packets, creating an identification that simplifies processing and reduces per-packet overhead. Note that changing IP addresses of packets requires recalculation of transport layer header checksums to avoid the packet to be dropped in network stacks. This is not repeatedly mentioned in the following description of the processing stages. Furthermore, we abbreviate IP address changes with (new source IP address, new destination IP address).

DEFAULT MAPPING APPROACH: TUNNELING. The simplest IP mapping method that many mobility management protocols apply is *tunneling*, which encapsulates the packet with the persistent IP addresses into a packet with the IP addresses for routing. Hence, the proxy does not require additional information for outgoing packets from the mobile node. Due to decapsulation of the packet, the proxy receives the actual packet, which can be forwarded. Accordingly, processing stages I and III add a second IP header for routing, while stages II and IV remove it. In stage I, the vehicle has to set the IP addresses of the initial header to (DA IP, CN IP) and add the outer routing header with addresses (MN IP, DA IP). This process constructs the inner packet for forwarding by the proxy, which is encapsulated in the outer IP header for routing. At Stage II, the proxy decapsulates the first header and stores the tuple (CN IP, transport protocol, src port, dst port) as connection identifier, mapping it to the MN IP address. Stage III has to look up the stored tuple for incoming packets, sets the internal header to (CN IP, MN IP) and creates the outer routing header with the addresses (DA IP, MN IP). Finally, in Stage IV, the mobile node decapsulates the packet and passes it to the upper layer.

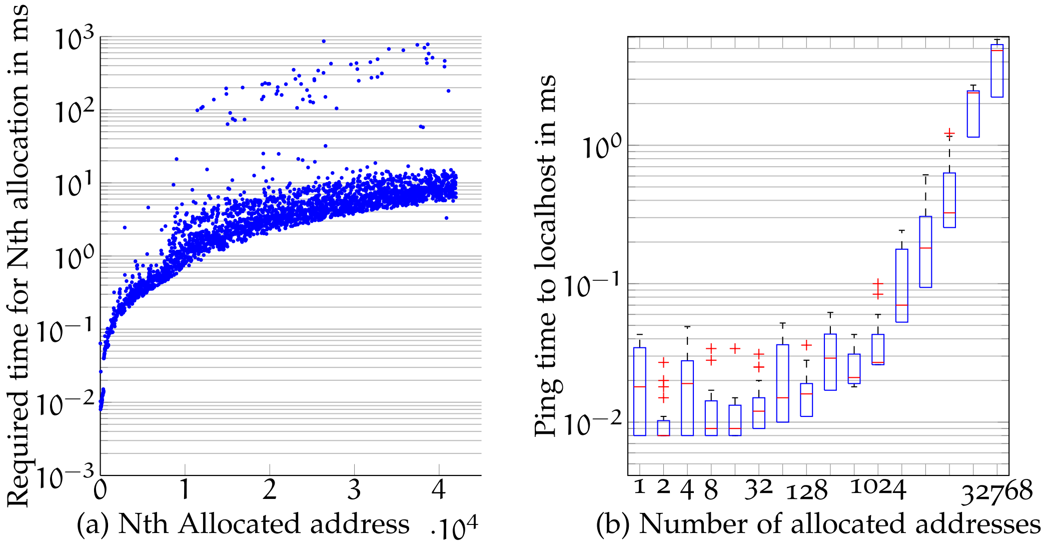
This default approach has two disadvantages. Firstly, tunneling causes a per-packet overhead of one IP header, i.e. 40 bytes, plus for many cases ensuring compatibility one UDP header of 8 bytes. Secondly, analyzing upper layer information to determine the forwarding destination introduces a considerable per-packet workload for the proxy machine. To address these issues, we employ the IPv6 address space in the following approach to simplify this process and reduce the overhead.

IMPROVED MAPPING APPROACH: FULL IPV6 ADDRESS MULTIPLEXING. IPv6 provides an address range to each node from which it can select one or more IP addresses for its own network interface, using IPv6 autoconfiguration [206]. To simplify the address translation process and reduce the packet overhead compared to tunneling, the Data Agent can use multiple IP addresses to multiplex data traffic, identifying the connection with its connection partners from the used Data Agent IP address. We design and discuss two alternatives ways for IP address multiplexing at Data Agents in the following, starting with the extreme approach, which uses an own Data Agent IP address for each connection: A **dedicated Data Agent IP address for each connection** enables induction of all connection parameters, i.e. IP addresses of the Mobile Agent, the Correspondent Node (CN) and identification of the connection itself. For the connection setup, the Mobile Agent couples the specific Data Agent IP address with the CN IP address. Accordingly, each packet arriving at the DA carrying this IP address is known to belong to the connection, with both its endpoints to which it must be forwarded. The approach does not rely on the connection's transport protocol, *reduces the packet analysis to the IP header* and renders any additional header obsolete, i.e. reducing the per-packet *overhead to zero*.

Unique DA IP addresses are dedicated to a single connection only and, therefore, can be employed for connection identification. This eliminates per-packet overhead and dependence on the transport layer.

However, the concept requires extensive allocation of Data Agent IP addresses, i.e. for each active connection one. Since unicast-address announcement requires some effort in the network, covering time-consuming duplicate address detection [206, 149, 53] and routing table updates of neighboring nodes, the Data Agent could make use of aggregated route allocation [221], announcing a complete IP address range to neighboring nodes. Even though this route aggregation is applied in practice within infrastructure route management [126] using Classless Inter-Domain Routing (CIDR) [74] or Open Shortest Path First (OSPF) [44, 152], there seems to exist no approach to announcing IPv6 address ranges from nodes themselves. This might be due to the fact, that address ranges are usually used to unify routing to multiple nodes within a sub-network rather than to address a single multi-homed node. However, imitating these routing control messages, or applying similar rules in Software Defined Networks (SDN), could provide an efficient way to gather each packet at the Data Agent that is addressed to the covered IP range.

Nevertheless, the Data Agent's IP stack still has to be prepared to enable handling of the packets from these addresses. To assess scalability for handling complete IP ranges in current systems, we allocate multiple IP addresses in a Linux system (Ubuntu 16.04) on a Desktop Intel i7-4790K machine with 8 GB RAM. Therefore, we measure the time required for consecutively adding single IP addresses and perform latency tests through randomly addressed pings to generated IP addresses of the local machine, as shown in Figure 38. Plot (a) shows the required time for the allocation of the N^{th} IP address at a node. In the test, we allocated 42.000 IP addresses consecutively, measuring the required time for each allocation. We observe two effects: Firstly, the minimum required time for allocation of a single IP address rises with the number of already existing IP addresses. Starting in average from about 0.01 ms for the first addresses, it rises to 1 ms around the 10.000^{th} address and 10 ms around the 40.000^{th} one.



(a) Nth Allocated address

(b) Number of allocated addresses

Figure 38: Time required for allocating IP addresses and ping latency tests to the local machine with n addresses

Secondly, and more severely, the variance of the required time rises significantly. In several cases, the process for allocating a single IP address, which is expected to be finished within less than a millisecond, required up to one second. Furthermore, beyond allocation of the 42.000 addresses, we faced severe system instabilities in Linux, perceivable from about 8000 addresses on. Experienced instabilities included system freezes and network interface breakdowns that could not be recovered without a system restart.

Systems cannot cope with the extensive amount of IP addresses today, rendering full IP multiplexing inefficient.

Occurring problems are also reflected in interface responsiveness, which we measure using pings to the configured IP addresses on the local host, as presented in Figure 38 (b). After allocating a number of addresses, we ping randomly selected ones from the local machine, measuring the network interface responsiveness of the node itself. We observe an exponential rise of ping times when increasing the number of allocated IP addresses. Till allocation of 32 IP addresses, there is no significant effect on ping times to local host. However, after allocating 256 and more addresses, the ping rises from usually less than 10 µs up to 300 µs in median and finally to 4.83 ms for 32768 addresses.

The identified problems in network management and the presented impact analysis of the number of IP addresses of a node on latency and stability reveal that current networks and systems cannot handle this extensive number of IP addresses. This inefficiency renders the approach infeasible for the present time and opens new questions in practical network management, which reach beyond the scope of this dissertation. However, after closing these gaps, the presented principle of *full IPv6 address multiplexing* can be applied to optimize future mobility management approaches. To cope with the shortcomings of today's systems, we present a hybrid mapping approach in the following.

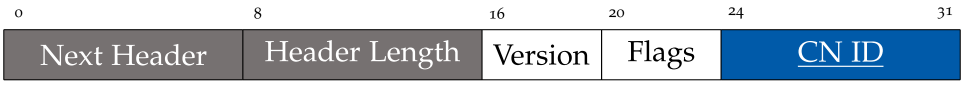


Figure 39: MoVeNet Identification header with optional correspondent node ID, used for data transfer with the hybrid mapping approach.

Using unique DA IP addresses per mobile node reduces the number of addresses significantly. To identify the forwarding destination, we introduce an additional 8 bit correspondent node ID.

HYBRID MAPPING APPROACH: NODE-DEPENDENT ADDRESS MULTIPLEXING. As the presented *full IPv6 address multiplexing* approach does not comply with current systems as required in *F-Req-3*. We propose a *hybrid approach* that allocates a **dedicated Data Agent IP address for each Mobile Agent**, significantly reducing overhead and processing complexity in comparison to tunneling but using significantly fewer IP addresses than the full IPv6 address multiplexing approach. In this hybrid approach, a Data Agent generates a single dedicated IP address for each Mobile Agent, serving for mobile node identification for all incoming and outgoing packets. An additional multiplexing for the IP mapping is still required to distinguish connections to different communication partners of the Mobile Agent. Therefore, the approach employs a correspondent node identifier (CN ID), sent along in the bridged communication between MA and DA. Transmitted CN IDs just have to be unique for each corresponding node with an active connection to the specific Mobile Agent, which is identified by the dedicated DA IP address. Therefore, the identifier can be short. Instead of using a 128 bit IP address to refer to the CN's IP address, as tunneling does, this approach sends along an 8 bit correspondent node identifier (CN ID), colored blue in Figure 39, to which the CN's IP address is mapped. This header causes a per-packet overhead of just one word, i.e. 4 bytes, and allows a Data Agent to maintain connections to up to 256 different corresponding nodes at the same time for each Mobile Agent. In the case of more required parallel connections, the Mobile Agent has to employ multiple Data Agents, as recommended in the architectural principles of our protocol MoVeNet.

During initialization, the mapping information between the CN ID and the CN IP is created at the Mobile Agent and transferred to the Data Agent, as detailed in Section 6.3.3, describing the initialization processes. We explain the four stages of the hybrid approach's IP mapping method in the following. In processing Stage I, the Mobile Agent looks up (or generates) the CN ID from the CN's IP address, replaces the addresses in the packet's IP header by (MA IP, DA IP) for packet routing and attaches the Identification header shown in Figure 39 to enable the mapping at the DA. In processing Stage II, the Data Agent looks up the CN IP from the attached CN ID and replaces the addresses in the IP header with (DA IP, CN IP) for packet forwarding, dropping the Identification header from the packet to convert the connection into a conventional one. For packets from the correspondent node, the Data Agent uses the DA IP at which the packet arrives to identify the corresponding Mobile Agent and its IP addresses. The approach profits from relying on the IP header only and not on transport layer information, as tunneling does. The DA looks up the CN ID from its database, rewrites the addresses in the IP header to (DA IP, MA IP) for routing and attaches the Identification header according to Figure 39 to the packet to enable the reverse mapping in the next stage at the Mobile Agent. Finally, the Mobile Agent can determine the CN IP from the attached

CN ID and revert the packet header, as required, for the upper layer protocol in Stage IV.

In addition, as long as a CN's established context at the Data Agent persists, packets to the dedicated DA IP address are forwarded via an active access network to the mobile node, enabling *return-routability*, an often-desired feature of mobility management. However, packet arrival still depends on the access network's and mobile node's firewalls. Note that, in the rare case in which two or more connections from a MA to the same CN via the same DA shall be routed via different access networks, the agents have to identify the connection through packet analysis (default today) including transport protocol and ports. Conclusively, in trade-off with a 4-byte per-packet overhead with respect to the full multiplexing approach as well as required packet analysis in the rare cases that multiple connections to the same CN should be routed differently, the hybrid approach solves the problem of extensive IP address initialization and, in extension, creates partial return-routability.

In the following, we present latency optimization considerations, which explain how Data Agents should be selected. Then we detail MoVeNet's management processes for system and connection initialization.

6.3.2 *Latency Optimization Considerations*

The Internet connection in a vehicle enables access to external data sources, which can complement local information sources of Advanced Driver Assistance Systems, contributing to the vehicle's safety. Low latency connections can provide information updates more recently, enhancing the system performance. Hence, a low connection latency is desirable at least for selected connections in the connected vehicle scenario. As we stated in Section 3.3, Data Agent placement close to the optimal route reduces the *connection latency*. To enable low latency connections, MoVeNet employs multiple Data Agents as proxies that can be selected individually for each data flow to be as close as desired to the optimal route. For non-critical connections, MoVeNet may aggregate multiple connections routes at a small set of Data Agents.

Furthermore, the selection of a Data Agent close to the Mobile Agent reduces the *handover latency* because signaling messages have to travel only a subset of the route [93]. This shortened signaling path enables a faster reaction on unexpected IP address changes and follows the concept of several Mobile IPv6 derivatives [129], especially hierarchical Mobile IPv6 (HMIPv6) [196, 163]. Even though we present location selection goals, an algorithm is out of the scope of this work. Such algorithms may be inspired from cloudlet selection algorithms [88, 219] or Software Defined Network controller placement algorithms [123, 220, 90], which select nodes to optimize route latency, failure tolerance and load balancing.

6.3.3 *Initialization Process*

As first initialization step of the MoVeNet protocol, a Mobile Node has to connect to a Control Agent, according to Figure **??**, in order to receive a unique identifier, the mobile node identification number (Node ID). It represents a persistent ID to refer to the mobile node with changing IP addresses within the distributed mobility management system. Since this process is employed only once, e.g. at device start up, we propose to use a secured remote procedure call (RPC). Using default mechanisms ensures up-to-date security and fixes, independent from the protocol specification and implementation. Next, to transmit data, a connection to a Data Agent has to be set up. Therefore, we propose two different mechanisms. The first mechanism (1) covers dedicated Data Agent selection and initialization, which can be used to optimize the connection characteristics, as detailed in the previous section. The second mechanism (2) complements the first regarding low-latency, using active Data Agents of the Mobile Agent without sending a request to the Control Agent. It is designed for short-lived and time-critical connection setup, as required for many use cases in the automotive scenario [64].

6.3.3.1 *Default Connection Initialization*

In default connection initialization, the CA selects a DA targeting low route latency between MN and CN.

MoVeNet's default connection initialization selects and prepares a Data Agent for transmission and sets up its communication stack states for packet modification, as shown in Figure 40 (1). Note that underlined parts are present only for the hybrid IPv6 multiplexing mapping approach. When the communication stack of the Mobile Agent identifies a new connection, it determines a DA ID (in hybrid mapping) and sends a Data Agent Initialization Request to the Control Agent, including DA ID, Node ID and the correspondent node's IP address, as illustrated in Figure 41. It is essential to let the Control and Data Agent know the target for forwarding, as MoVeNet replaces the original destination IP address for packet routing. Next, the Control Agent determines an appropriate Data Agent for this connection, taking into account its location, which affects the round trip time between the mobile node and the correspondent node due to packet detours and also the handover delay, as stated in Section 6.3.2, addressing *NF-Req-1*. It initializes a Data Agent using a secured remote procedure call (RPC), transmitting CN ID, Node ID, CN IP and all active MN IP addresses to the DA. The Control Agent keeps this connection alive for Data Agent maintenance, including failure monitoring [94, 82, 174] and later MA IP updates. If not already existing, the Data Agent allocates an DA IP address (or a range of addresses in case of full IPv6 multiplexing) and prepares the connection context, mapping Node ID and CN ID to the allocated IP address. Connection context establishment at the Data Agent covers allocation of a new IP address (range for full IPv6 multiplexing) for the Mobile Node, if not already allocated, and setting up forwarding rules for the new connection. Finalization of this setup process is acknowledged to the CA and, in turn, to the MA, which can consecutively start to send data to the correspondent node via the Data Agent. In the meantime, MoVeNet caches arriving packets at the MA.

The procedure initializes Data Agents and enables their optimized selection for each connection. However, it creates a certain delay till the actual data transmis-

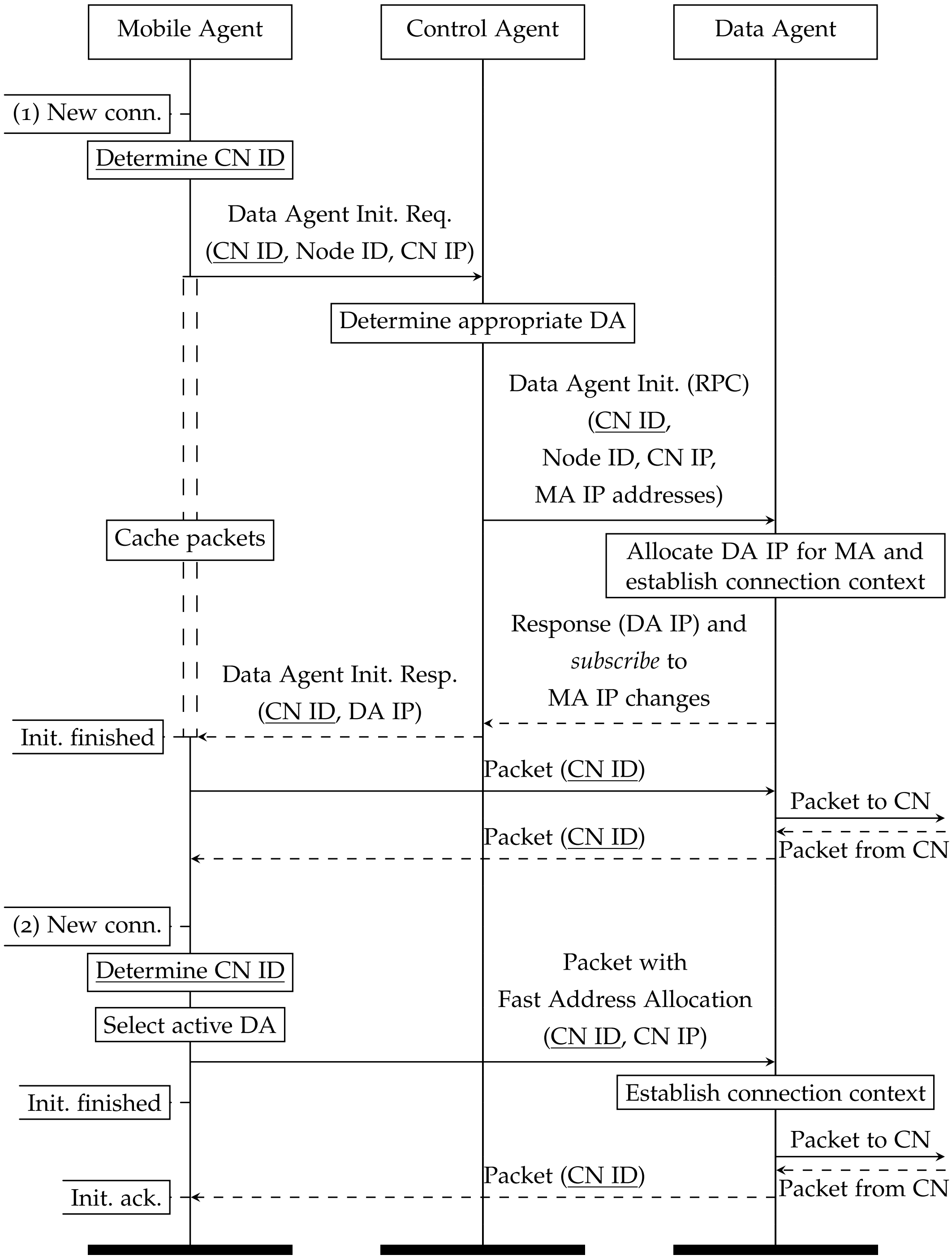


Figure 40: MoVeNet connection initialization using (1) the default mechanism, initializing a new Data Agent at an appropriate location and (2) the piggybacking-based Fast Address Allocation mechanism, reducing initialization delay due to selecting a Data Agent from already active ones. Underlined packet fields are required in the hybrid approach but not in the full multiplexing.

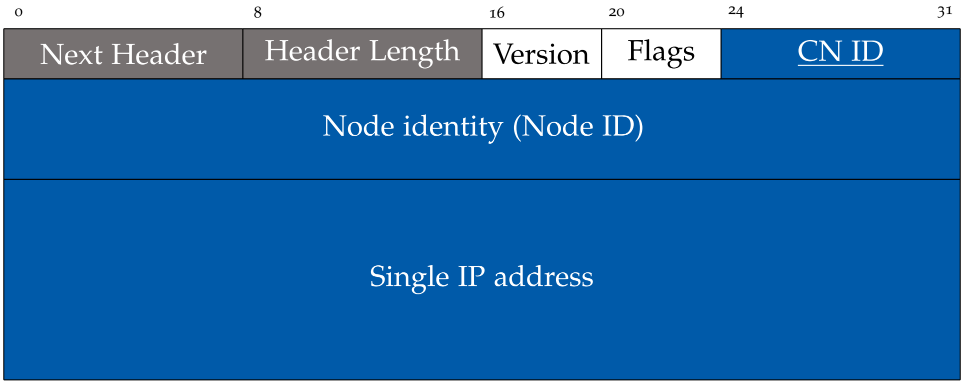


Figure 41: MoVeNet Identification header of a Data Agent Initialization request, covering correspondent node ID (for hybrid mapping approach only), Node ID and Single IP address field, carrying the correspondent node's IP address.

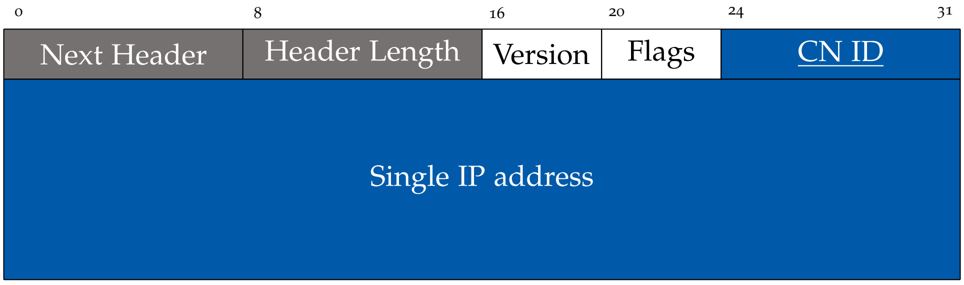


Figure 42: MoVeNet Identification header with correspondent node ID (for hybrid mapping approach only) and Single IP address field. For Data Agent Initialization Response, it carries the allocated DA IP address. For Fast Address Allocation, it carries the correspondent node's IP address (CN IP). In this case, the MA is already identified from the used DA IP address.

sion can start. Therefore, we propose a second mechanism in the following, which reduces the start-up delay.

6.3.3.2 *Fast Piggybacking-based Connection Initialization*

The complementary connection initialization mechanism Fast Address Allocation (FAA) skips requesting the Control Agent for a Data Agent, instantly sending packets to a known Data Agent, employing the Identification header to piggyback all information for setting up the connection to a correspondent node. As a prerequisite, the Mobile Agent must already maintain the Data Agent, which, in return, knows the Mobile Agent's context from previous or running connections, established by the default connection initialization mechanism. In this case, optimized Data Agent selection is restricted to known ones and must be accomplished with the potentially limited knowledge of the Mobile Agent. We consider this approach as beneficial, firstly, for time-critical connections, which require timely transmission and, secondly, in the case that the Control Agent is not responding, e.g. due to node failure.

FAA sends packets without prior announcement to a known DA, piggybacking all required information to set up the IP mapping context.

To initialize the connection, in every outgoing packet from the Mobile Agent, the employed MoVeNet Fast Address Allocation (FAA) header covers the CN IDs and the CN IP address, using the single IP option as shown in Figure 42. With this information, the Data Agent is able to establish the IP address mapping context for the connection. Receiving a packet including the CN ID, acknowledges correct connection setup to the Mobile Agent, which, in turn, stops attaching the Fast Address Allocation fields in the MoVeNet Identification header to outgoing packets. The new mechanism eliminates the detour via the Control Agent and sends packets instantly to the Data Agent without creating any further initialization delay, addressing *NF-Req-2* attaching all information required to set up the connection context.

6.3.4 *IP Table Synchronization*

All MoVeNet Agents maintain a table of currently valid IP addresses of the mobile node. Data Agents and the Mobile Agent use this set of IP addresses to select the desired route for running connections. To ensure proper protocol operation, the table of valid IP addresses of the mobile node has to be up-to-date at the agents. Synchronization delays either reduce the efficiency of the transmission because valid routes are not employed or, even worse, late invalidation of IP addresses leads to sending packets to dead ends, resulting in avoidable packet loss. Therefore, it is important to synchronize the IP tables quickly. For this reason, MoVeNet offers two mechanisms: (1) a default mechanism reducing the signaling overhead and (2) an optional complementing low-latency mechanism, both visualized in Figure 43.

6.3.4.1 *Publish-Subscribe-Based IP Table Synchronization*

Our default mechanism for IP table synchronization uses the Control Agent as a publish-subscribe broker to distribute signaling information from the mobile node to all subscribed Data Agents, as illustrated in Figure 43 (1). The Mobile Node informs its Control Agent about the IP address update, using the MoVeNet IP Synchronization header fields, shown in Figure 44. Consecutively, the Control Agent synchronizes the routing tables of the Data Agents, which are subscribed to updates in the context of the Mobile Agent, using remote procedure calls (RPC), if possible via the kept-alive connections from Data Agent initialization. As soon as synchronization of all subscribed Data Agents is completed, the Control Agent answers to the Mobile Agent with a synchronization acknowledgment, releasing new announced IP addresses for operation.

The default route announcement is sent to the CA, which acts as a publish-subscribe broker and distributes the information to DAs to minimize signaling overhead.

This publish-subscribe based mechanism for IP table synchronization reduces signaling overhead, addressing *NF-Req-3*, compared to informing each Data Agent or even each connection on its own, which is the default in most other protocols [185]. However, the detour via the Control Agent to reach each Data Agent introduces additional latency. Hence, the mechanism is designed for connections, for which a few hundred milliseconds extra handover delay are not relevant. Note that due to multi-homing and the applied make-before-break principle, i.e. keep using the previous link until the new network link is ready to transport data, this handover delay does not necessarily denote a transmission pause but only a de-

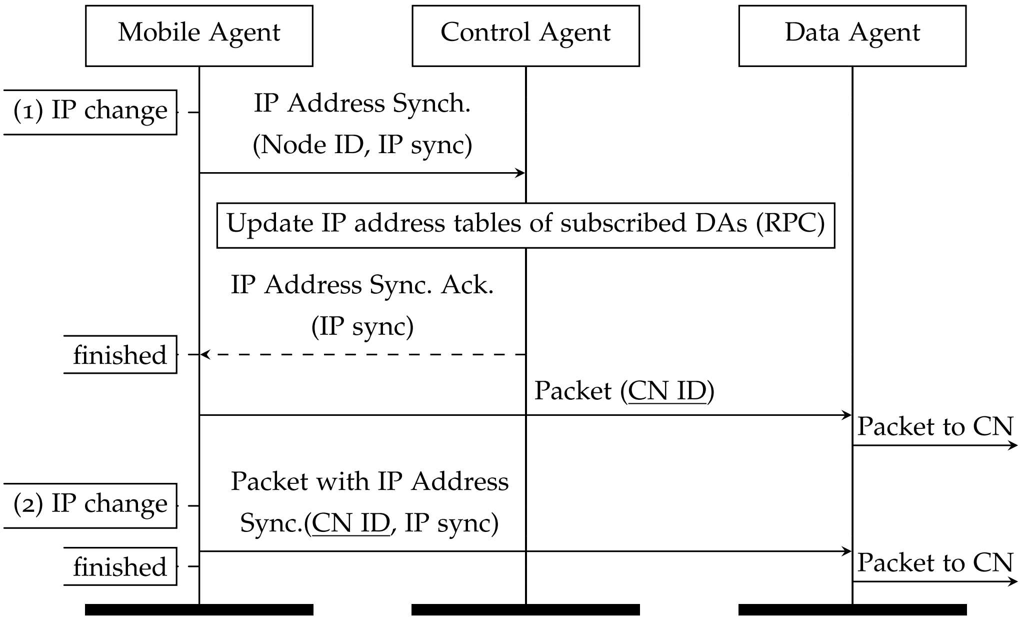


Figure 43: IP address synchronization mechanisms. (1) Default IP address update using a publish-subscribe mechanism with Control Agent as a broker. (2) Optional fast piggybacking-based IP address update.

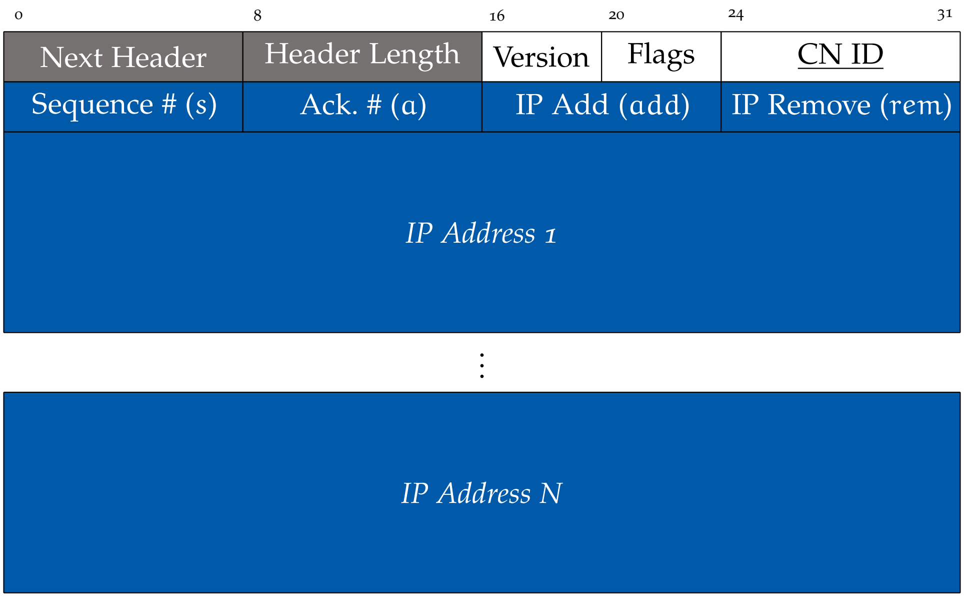
layed use of newly available networks. However, in the case that no redundant second access network is available, the make-before-break principle is not applicable. Hence, several non-delay-tolerant connections rely on fast network continuity. To address this issue, we designed an optional complementary second synchronization mechanism, introduced in the following, which speeds up informing of Data Agents about new IP addresses.

The complementary IP table synchronization attaches the information to data packets to the DA. Thus, they can be sent via the new route without prior announcement, achieving instant handovers.

6.3.4.2 *Piggybacking-Based IP Table Synchronization*

We complement our first mechanism with an optional piggybacking-based IP update mechanism, which allows MoVeNet to use new routes instantly without prior announcement, closing the handover delay gap of the first mechanism and addressing *NF-Req-2*. It is illustrated in Figure 43 (2), employing the optional MoVeNet IP Synchronization header fields, as shown in Figure 44, to attach IP address changes to every data packet from the mobile node to the Data Agent. As a result, the Mobile Node may send packets via new routes as soon as they are available, even before synchronization of Data Agent IP tables because all required information to accept and forward the packet is covered in the attached IP Synchronization header fields.

With this information, the Data Agent can update its IP table. To acknowledge correct reception of the IP update, it attaches the MoVeNet IP Synchronization header option as well. As soon as the synchronization is acknowledged, i.e. at least after one round trip time between Mobile Agent and Data Agent, the Mobile



Next Header	Header Length	Version	Flags	CN ID
Sequence # (s)	Ack. # (a)	IP Add (add)		IP Remove (rem)
IP Address 1				
⋮				
IP Address N				

Figure 44: MoVeNet Identification header with optional piggybacking IP Synchronization header fields marked blue. Sending IP updates along with the packet speeds up the synchronization in trade-off for higher overhead.

Agent stops attaching this information to packets. Hence, the decision whether to use the mechanism or not, represents a trade-off between handover latency and overhead. We propose to use it for time-critical transmissions only. Note that IP synchronization affects all connections of the addressed Data Agent. Thus, applying the piggybacking-based fast IP Address updates to the most critical ones, all other connections profit from this early update as well; however not from the first packet but after the arrival of the information at the Data Agent, respectively the acknowledgement of the Mobile Agent. Hence, the new mechanism eliminates additional handover delays completely, allowing the Mobile Agent to send packets instantly via new connections to the Data Agent, without prior announcement.

6.3.4.3 MoVeNet *IP Synchronization header*

The employed MoVeNet IP Synchronization mechanism enables synchronization even during concurrent modification of IP tables at both communication partners. The header fields consist of one mandatory word, i.e. 4 bytes, as depicted in Figure 44. The first and second bytes in the header are a sequence number s and an acknowledgment number a, showing the state of synchronization of the communication partner. The third and fourth are counters that determine the number of attached IP addresses to be added add to the IP table T and the number of IP addresses to be removed rem from the IP table T of the communication partner.

The MoVeNet *synchronization mechanism enables multiple and even concurrent modifications of IP tables at both ends.*

To synchronize the IP address tables T, the designed mechanism employs (1) local sequence numbers s_l and s_r, (2) a local acknowledgment number a_l and (3) a fifo-list L containing unacknowledged IP address changes, flagged if they are added or removed. The header fields s and a of outgoing packets in an IP Synchronization header are filled with the local numbers s_l and a_l. The counters add and rem are set according to the entries of the fifo-list L. First, all IP addresses to be added are attached to the packet, followed by the IP addresses to be removed from the communication partner's IP table T. The synchronization actions are as follows:

- *Send IP address changes:* For each IP address change in the local IP table T, a corresponding entry is added to the fifo-list L and the sequence number is increased by one $s_l = s_l + 1$.
- *Send acknowledgment:* IP address changes of the communication partner can be detected by comparing the received sequence number s to the received acknowledgment number a, calculating the number of changes $c = s - a$. To avoid handling of duplicate received IP synchronization headers, the received sequence number is set $s_r = s$. The following update procedure is triggered only in case of $s > s_r$. The received changes are applied to the local IP table T. To acknowledge correct reception and to provide independence of own changes from the communication partner's ones, the local sequence and acknowledgment numbers both are increased by the number of received IP address changes $s_l = s_l + c$, $a_l = a_l + c$. Adding c to both numbers decouples received updates from local changes for the two communication partners, enabling concurrent modification of the local IP tables T.
- *Receive acknowledgment:* New acknowledgments can be identified by comparing the local acknowledgment number to the received one, calculating the count as $c_a = a - a_l$. To signalize that an acknowledgment has been received, the local acknowledgment number is set to the received one $a_l = a$ and c_a items are dropped from the fifo-list L of unacknowledged IP changes.

If the piggybacking mechanism is activated for a connection, the MoVeNet IP Synchronization header is attached to IP packets due to three triggering conditions: (1) to send IP changes if the fifo-list L is not empty, (2) to acknowledge IP changes if a packet is received with $s > a$ or (3) to confirm acknowledgment reception in order to make the communication partner stop sending acknowledgments if a packet is received with $a \neq a_l$. In contrast to the simple update-ack mechanisms of Mobile IPv6, HIPv2 or SHIM6, this mechanism facilitates MoVeNet to announce even parallel IP changes at any point in time and not only consecutively, one after another. This advanced synchronization improves the protocol's flexibility in handling environmental changes according to *F-Req-2* and reduces handover latency complying to *NF-Req-2*.

6.3.4.4 *Control and Data Agent Synchronization*

Applying the MoVeNet header for IP address updates from the Mobile Agent to the Control and Data Agents has two major purposes: Firstly, it can be transmitted easily from bypassing sockets. As connections are managed through MoVeNet, a signaling communication using higher layer services would be treated from the system. In the case of failures, the system would not be able to recover itself because signaling is blocked from the non-working managed link, creating a system deadlock. Thus, the MoVeNet header attached as IP extension represents an independent mechanism, ensuring system robustness according to *NF-Req-4*. Secondly,

Table 13: Specification of the Identification header flags. The left table shows the assignment of message types to flag values. The right table contains the bitmasks used to determine the presence of header fields.

Header Field	Flags
CN ID	0001
Node ID	0010
Single IP	0100
IP Synchr.	1000

Message Type	Receiver	Mask
Data	MA/DA	xxx1
Data Agent Init. Request	CA	0111
Data Agent Init. Ack.	MA	0101
Fast Address Allocation	DA	x101
IP Address Synchronization	MA/CA/DA	1xxx

the approach reduces the overhead and can be attached to critical packets, eliminating delays to satisfy *NF-Req-2*.

However, Control and Data Agents are both connected to the Internet with wired interfaces, which are, firstly, not managed through MoVeNet and, secondly, less restricted in terms of throughput. Hence, these Agents are free to use default synchronization mechanisms, whose efficiency, security and robustness are proven and maintained from the operating systems. We apply remote procedure calls (RPC) to accomplish the communication, keeping alive the used connection to omit the connection setup, reducing the synchronization delay. We propose to use secured remote procedure calls, allowing a reliable and efficient transfer. As synchronization data structure, Conflict-Free Replicated Data Types [22, 192] may be applied, which offer the advantage of strong eventual consistency [51], mitigating errors from synchronization and concurrent updates.

6.3.5 *Identification Header*

The full MoVeNet Identification header is shown in Figure 45, consisting of one mandatory word and several optional ones. It starts with the fields *Next Header* and *Header Length*, dictated from the IPv6 extension header specification [50]. They are followed by the 4-bit fields *Version*, flags *Flags* and the 8-bit correspondent node ID (CN ID), required for the hybrid IP mapping approach only. The flags indicate the presence the optional header fields CN ID, Node ID, Single IP, IP Synchronization, as presented in the previous sections and shown in Table 13 left. From their presence, the message receiver can identify the type of message, as presented in Table 13 right. Hence, all data packets passing the handover-enabled data bridge contain the CN ID (since using the hybrid mapping approach). Data Agent Initialization messages contain a Single IP field, which is used to signalize the intended IP mapping. Furthermore, the Node ID as header field is required only for identification at the Control Agent and is used to refer to the Mobile Agent from the backend entities, i.e. Control and Data Agents. Finally, the IP address synchronization mechanisms set the fourth flag, signalizing the presence of the corresponding header fields.

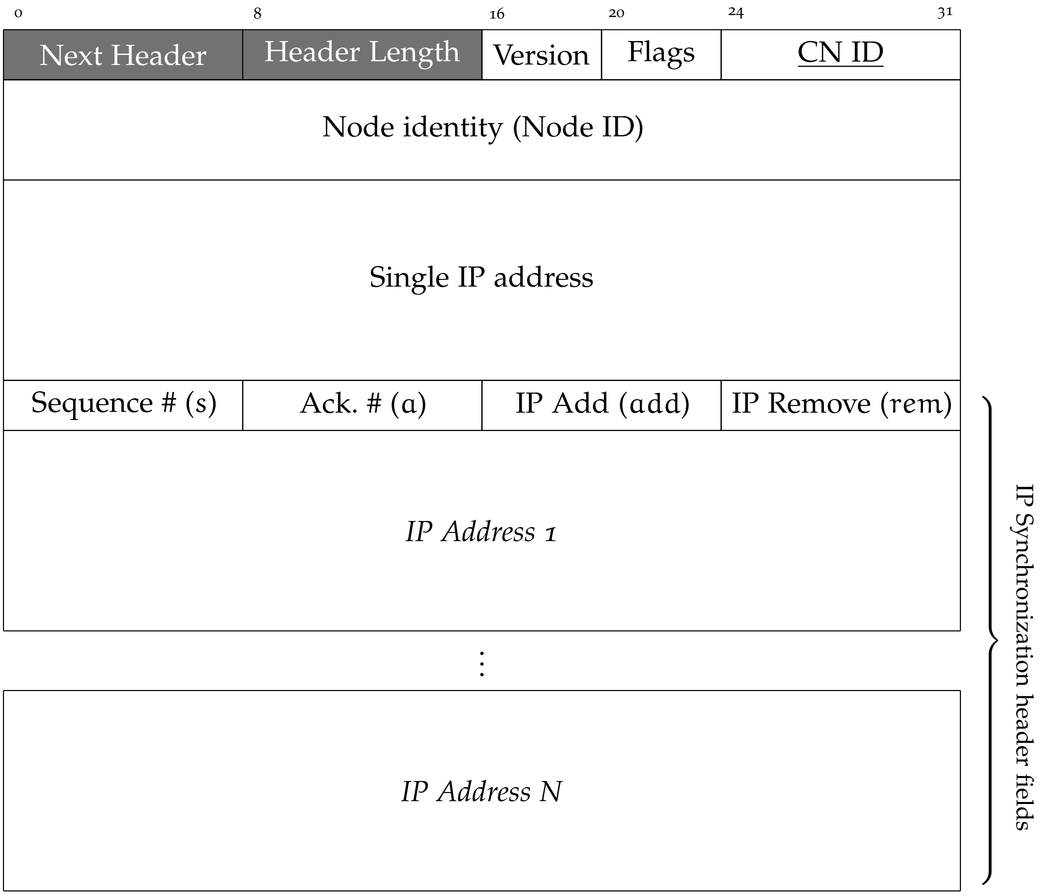


Figure 45: MoVeNet full Identification header with all optional fields marked blue.

6.3.6 *Event-Based Retransmission Trigger*

MoVeNet introduces a new mechanism to let reliable network protocols continue transmission instantly after network interruptions due to resending the last passed packets as an external trigger, improving transmission efficiency in environments with short-lived network connectivity.

The event-based retransmission trigger addresses TCP's inefficient transmission recovery in sparse network environments. Transmission is triggered externally as soon as the connection is up again.

When no Internet access is available for more than a second, reliable protocols, as TCP or SCTP, pause transmission until a timeout triggers retransmission to probe the connection. Each time the probing fails, the timeout duration is doubled until the connection times out completely. This mechanism of TCP as well as SCTP works independently from other layers, following the principle of separation of ISO/OSI layers.

In a harsh environment of short-lived link connectivity, the absence of access networks for several seconds and intentionally paused transmission from control mechanisms, this may lead to severe performance degradation and unused resources. We provide an example in Figure 46, showing an extreme case for better understanding. In the example, the exponentially rising retransmission trigger timeouts, illustrated as blue arrows, attempt packet retransmission in situations without connectivity. The long retransmission timeouts waste resources, when net-

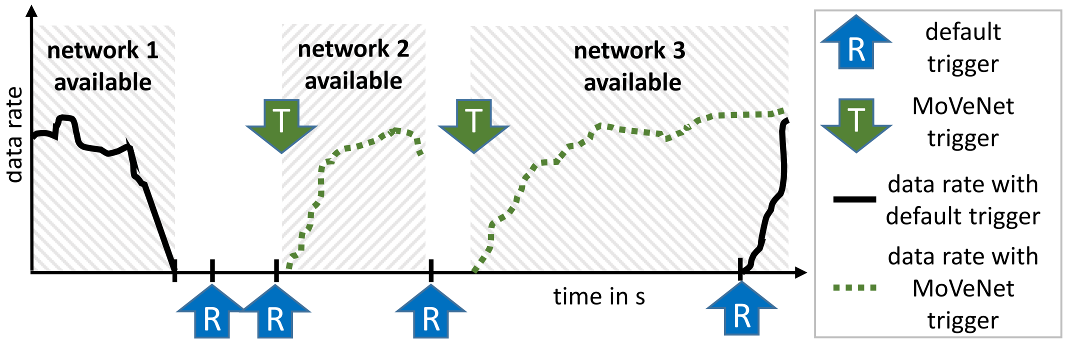


Figure 46: Extreme case example showing MoVeNet's Advanced Retransmission Trigger. Default retransmission triggers fail, probing the network during periods without network access. In contrast, the MoVeNet Trigger takes information from its managed lower layer into account to continue transmission as soon as possible.

works get in reach, leading to unnecessary long transmission pauses. To address this issue, MoVeNet employs its network layer knowledge to trigger packet retransmission externally, visualized by the green trigger arrows in Figure 46. The mechanism caches the last seen incoming and outgoing packet of each connection in the Mobile Agent. As soon as a new access network is available, the Mobile Agent repeats passing the packets, serving as an external trigger for transport layer protocols to recover from transmission pauses, as shown with the green dotted line. This simple mechanism can create significant performance benefits in continuing paused transmissions.

6.3.7 *Security Considerations*

Employing IP extension headers, as in the presented protocol MoVeNet, is considered as a security risk by Gont [80]. Especially mobility management protocols are subject to special threads on IP binding updates, legitimating re-routing and injection of packets. To encounter these security issues, different proposals exist. Firstly, only Cryptographically Generated Addresses [12] should be used to ensure that the source IP address belongs to the right client. However, this approach cannot authenticate additional header fields of MoVeNet. Hence, whenever the MoVeNet Identification Header transports sensitive information, i.e. for IP synchronization, we propose using IPv6 Authentication headers [108] encapsulating the MoVeNet header to ensure integrity and authenticity.

An alternative approach is using the Encapsulating Security Payload (ESP) [109], which encrypts the whole packet in IPsec fashion, as considered for mobility management by Santa et al. [188]. However, this requires higher computational efforts and leads to significant protocol overhead. A second alternative approach is embedding a Keyed-Hash Message Authentication Code (HMAC) [121, 210] as employed in the Host Identity Protocol (HIPv2) [151] or Multipath TCP [73]. It provides the advantage of a smaller overhead, embedded as a protocol header field of the mobility management protocol. However, being integrated into the protocol itself, the security mechanism, and thus the whole protocol, has to stay up-to-date in its specification and all its specific implementations within the fast changing security sector. Therefore, we prefer to rely on default mechanisms, which are maintained by the operating system, decoupling security from the protocol tasks.

Furthermore, MoVeNet employs remote procedure calls as default service communication to synchronize the backend entities, i.e. the Control and Data Agents. We propose to use Transport Layer Security (TLS) [175], applied for most secure communication on the Internet.

6.3.8 *Robustness Considerations*

Due to MoVeNet*'s complementary management mechanisms, the MA has alternative ways for system control in case of node failure of the CA and a DA. This creates substantial robustness.*

The distributed design of MoVeNet and redundant mechanisms create an intrinsic system robustness against node failures, as required from *NF-Req-4*. On failure of a Data Agent, only the subset of active connections using this DA is affected. Since the Control Agent is in constant connection with Data Agents, we propose to use a monitoring at this node. In case of a non-responding Data Agent, the Control Agent may instantly create a new one with similar characteristics for the Mobile Agent and forward the new Data Agent IP address. In the meantime, the Mobile Agent may use any other known Data Agent to re-initiate the corrupted connections.

In the case of a Control Agent failure, no active connections are affected. Data Agents keep working even without the Control Agent. However, new connections have to be initialized using Fast Address Allocation and IP address synchronization must use the piggybacking-based mechanism to contact the Data Agents directly. In parallel, the Mobile Agent should initialize a new Control Agent and new Data Agents to recover seamlessly from CA failure.

6.3.9 *Further Overhead Reduction*

The piggybacking-based control mechanisms, as presented in this chapter, attach extra information to each outgoing data packet until the corresponding synchronization or initialization is acknowledged. Since the packet contains sensitive information, the authenticity of those packets must be ensured, using the abovementioned mechanisms. Both add substantial additional overhead to the packets. However, the update is already achieved with the first arriving packet, containing this information. Each consecutive piggybacked packet does not contribute to the function of the protocol and serves only as a backup for the previous ones. To reduce the overhead according to *NF-Req-3* induced from those mechanisms, a specific implementation of MoVeNet may attach this information only to the first, or the first few packets sent out to the Data Agent. This method decreases the reliability of the protocol, since each packet, which arrives without piggybacked information before the update is accomplished must be dropped from the Data Agent for practical or security reasons. However, the probability that one of the first packets arrive is very high. The used strategy for omitting this information may consider the criticality of the data connection and remains open to the specific protocol implementation.

6.4 ANALYTICAL EVALUATION

In the following, we discuss the design decisions of MoVeNet with resulting system properties and compare them to the targeted set of functional (F-Req) and non-functional (NF-Req) requirements. Consecutively, we detail the handover latency composition in a structural analysis.

6.4.1 *Design Implications*

Managing traffic on the network layer based on IPv6, MoVeNet can handle all data traffic of conventional nodes. For legacy IPv4 data traffic or IPv4 based access networks, MoVeNet may employ IP 6to4, respectively 4to6, tunnels between the Mobile Agent and the Data Agent. Since the 'waist' of the modern Internet is IP, MoVeNet is capable of handling 'all' data traffic of clients to the Internet, satisfying requirement *F-Req-1*. The multi-homing concept of MoVeNet, employing a dynamic mapping of individual connections to available networks, allows a client to distribute data traffic as desired. This is required to enable execution of data plans, as presented in Chapter 4, and satisfies the requirement of high routing flexibility *F-Req-2*.

MoVeNet communicates via proxy servers that hide the protocol from communication partners. The proxies hide the protocol operation and create compatibility to communicate to every node in the Internet, which employs a conventional network stack, fulfilling requirement *F-Req-3*. The applied distributed mobility management principle allows MoVeNet to select proxy servers reasonably per connection, firstly, to use close to optimal data routes, reducing round trip times according to requirement *NF-Req-1* and, secondly, to minimize handover latency, accounting for requirement *NF-Req-2*. Latency for handover and connection setup can selectively be improved further using the presented alternative piggybacking-based signaling mechanisms. In contrast, the default signaling mechanisms target and satisfy the low-overhead requirement *NF-Req-3*, while the improved mapping concept, employing single-interface IPv6 address multiplexing, reduces the per-packet overhead to 4 bytes for the presented hybrid mapping approach; respectively to zero bytes for the full multiplexing mapping approach, which may be applicable to future systems. The distributed system structure, employing redundant proxy servers, coupled with the integration of alternative signaling mechanisms that can deal with the failure of other nodes, creates strong system robustness as well as system scalability, addressing and satisfying *NF-Req-4*.

6.4.2 *Handover Delay Analysis*

To assess handover delays, we compare the time from the handover trigger till the first packet has arrived via the new route. Absolute values from simulation show trends and effects, which strongly rely on the actual (simulated) underlying network environment. Therefore, we first formally derive the handover delays from the scenario subroutes. The handover consists of two stages, firstly, signaling,

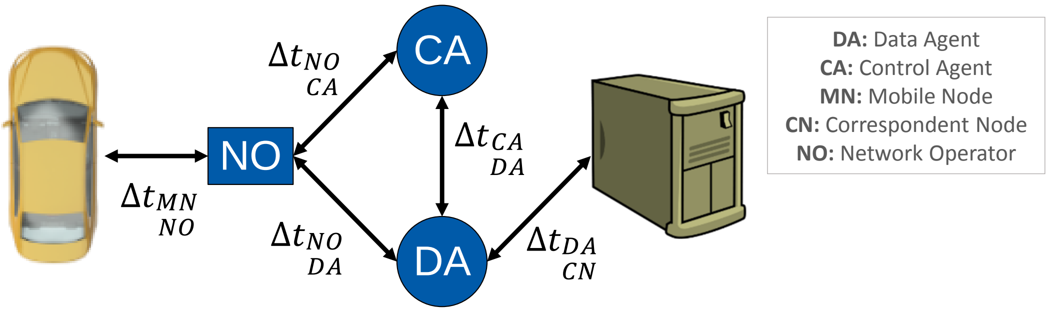

$\Delta t_{\substack{NO\\CA}}$
CA
NO
$\Delta t_{\substack{CA\\DA}}$
$\Delta t_{\substack{MN\\NO}}$
$\Delta t_{\substack{NO\\DA}}$
DA
$\Delta t_{\substack{DA\\CN}}$
DA: Data Agent
CA: Control Agent
MN: Mobile Node
CN: Correspondent Node
NO: Network Operator

Figure 47: Delays between entities of MoVeNet

which informs all involved nodes about the route update, and, secondly, routing, which covers passing of packets via the new route.

We formally describe the line delays in the MoVeNet ecosystem, covering the mobile node, the network operator's point-of-presence (NO), the Control Agent (CA), the Data Agent (DA) and the correspondent node (CN). Between the nodes, there exist link delays, as labeled in Figure 47 with Δt, considering symmetric links, i.e. similar delays in both directions of the links.

As summarized in Equation 6.1, each handover delay Δt_{HO} covers the delays for signaling and the final packet transfer. In the default mechanism, signaling consists of the path to the DAs via the wireless link and the CA, while data takes the direct route to the Data Agent. As soon as the DA receives the signaling information, it may use the announced link for forwarding data to the MN. The handover is finished from the point of view of the Data Agent. However, for data from MN via the DA to a correspondent node, the MN has to wait for an acknowledgment before sending data to ensure that the DA is ready to accept and forward data via the new route. We express the additional delay for the MN in the Equation 6.1 using the Identity operator $\mathbb{I}(x = y)$ which is 1 if $x = y$, i.e. adds this delay factor to the handover for the Mobile Node's point of view, else 0. Employing the triangle inequality to simplify the equation, giving a lower bound of the delay by neglecting the routing detour via the Control Agent, we receive Equation 6.2.

$$\Delta t_{HO}^{default} = \underbrace{(\Delta t_{\substack{MN\\NO}} + \Delta t_{\substack{NO\\CA}} + \Delta t_{\substack{CA\\DA}})}_{\text{signaling}} \cdot (1 + \underbrace{\mathbb{I}(node = MN)}_{\text{signaling ack.}}) + \underbrace{\Delta t_{\substack{MN\\NO}} + \Delta t_{\substack{NO\\DA}}}_{\text{data transfer}} \tag{6.1}$$

$$\Delta t_{HO}^{simple} = (\Delta t_{\substack{MN\\NO}} + \Delta t_{\substack{NO\\DA}}) \cdot (2 + \mathbb{I}(node = MN)) \tag{6.2}$$

In contrast to most other mobility management protocols, MoVeNet separates the control from the data plane, i.e. from data forwarding proxies. The default handover mechanism sends signaling information to the CA, acting as a publish-subscribe broker, which distributes it to subscribed DAs, reducing overhead and simplifying control. In architectures without control and data plane separation, e.g. Mobile IPv6, the DAs can be seen as integrated into the CA, reducing the delay $\Delta t_{\substack{NO\\CA}} + \Delta t_{\substack{CA\\DA}}$ to $\Delta t_{\substack{NO\\DA}}$, equal to the lower bound of MoVeNet, as presented in Equation 6.2.

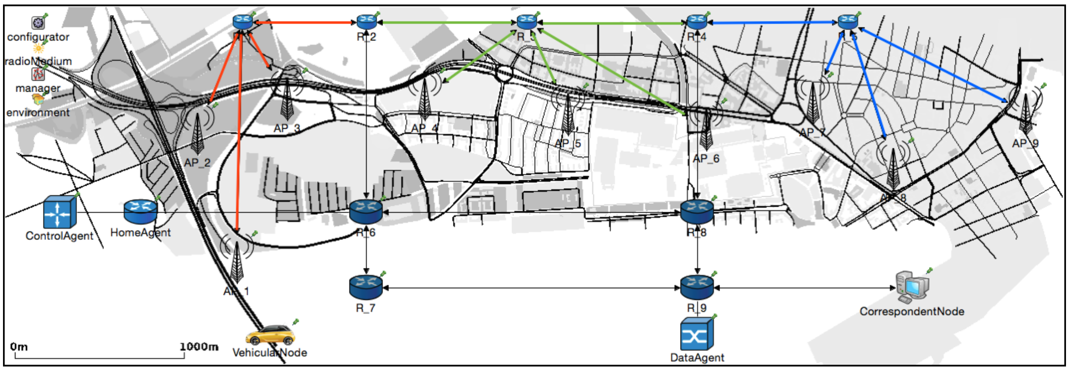


Figure 48: Handover Delay Simulation Scenario

Furthermore, the art of reducing handover delays employs hiding parts of it in other processes. For example, Proxy Mobile IPv6 [83] uses data link layer signaling to identify the mobile node. The access router informs the proxy about the new link location and IP address, hiding the signaling latency as well as data transfer till the access router in the delay of the network layer connection setup [128]. However, this is feasible only in a completely managed network and, thus, favored for intra-operator handovers.

In contrast, piggybacking mechanisms of MoVeNet eliminates the signaling delay due to hiding it in the data transfer delay, sending all required information to process the packet along with data. An announcement is no longer required for data sent from the MN to a correspondent node, giving instant access to new routes for this data transfer direction and reducing the delay to data route latency. In contrast, for data sent to the mobile node, the DA needs a route announcement, thus doubling the delay until the first packet arrives at the destination, the mobile node. This results in a handover delay Δt^{fast}_{HO} as defined in Equation 6.3, employing the identity operator to model the data direction dependent handover delay. Comparing the delay to that of Mobile IPv6, we identify a delay improvement of 2/3 for data from the mobile node.

MoVeNet's fast piggybacking-based IP synchronization mechanism reduces handover delay for the MN by at least 2/3.

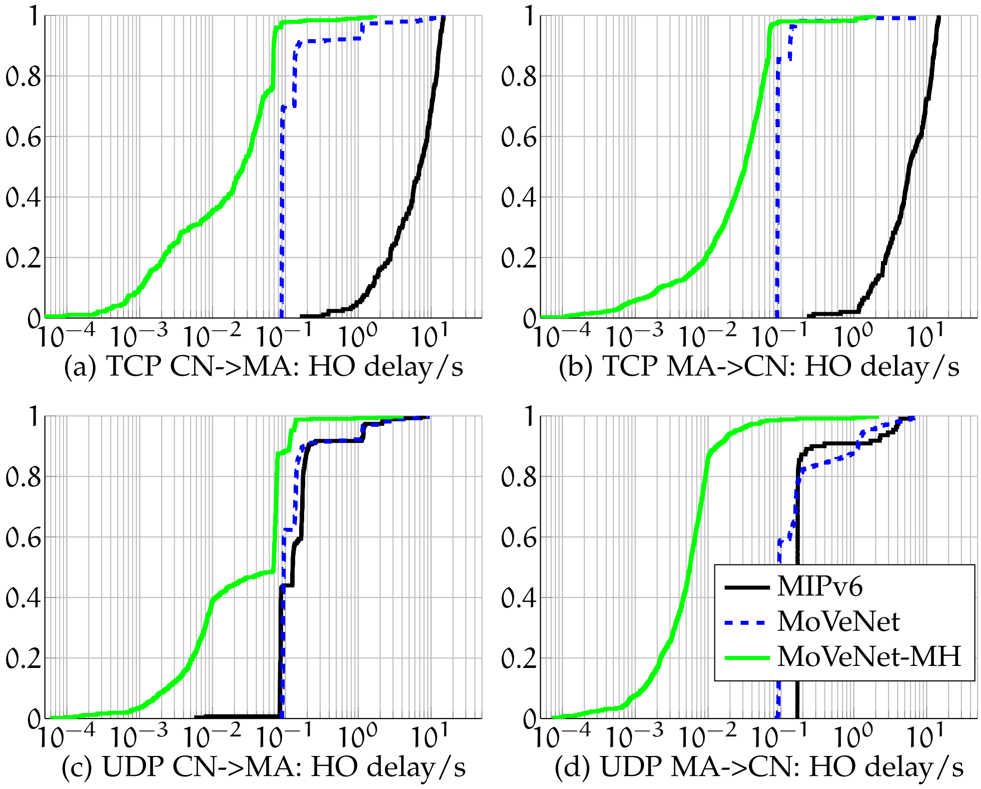
$$\Delta t^{fast}_{HO} = \cdot \underbrace{\left(\Delta t_{\substack{MN\\NO}} + \Delta t_{\substack{NO\\DA}}\right)}_{\text{data transfer to DA with signaling}} \cdot \left(1 + \underbrace{\mathbb{I}(node = DA)}_{\text{signaling ack.}}\right) \tag{6.3}$$

6.5 SIMULATIVE EVALUATION

In the following, we describe our simulation setup and evaluate the handover delay as well as the effects of MoVeNet's retransmission trigger.

6.5.1 *Simulation Setup*

We simulated MoVeNet with a single-homed and with a multi-homed node using Oment++ 4.6 with the INET 3.0 framework. For reference, we additionally simu-



(a) TCP CN->MA: HO delay/s

(b) TCP MA->CN: HO delay/s

(c) UDP CN->MA: HO delay/s

(d) UDP MA->CN: HO delay/s

Figure 49: Cumulative distribution of handover(HO) delay in s

late the Mobile IPv6 implementation *xmipv6* available for single-homed clients in the INET framework. To integrate vehicle mobility, we use SUMO 0.26 (Simulation of Urban MObility). We created a mixed scenario, covering motorway, suburban and urban areas, letting a single vehicle pass the track from Frankfurt Niederrad to Frankfurt Alte Oper (old opera). Along the track, we locate 9 WiFi access networks (802.11g) with an extended range of about 500 meters, as illustrated in Figure 48, covering the track completely. They are connected to ISP networks, colored in the graph, which consists of routers R1 to R5 (on the top of the picture), interconnected via links with 10 ms latency to model intra-ISP processes. Agents and ISP networks are interconnected with the backbone network, defined by R6 to R9, using links with 1ms latency. We define four runs, differing firstly in the transport protocol, either using a single UDP or TCP flow and, secondly, in the direction of the communication, either from the mobile node to the server or in the opposite direction. We compare the performance of three mobile nodes: (1) a single-homed node using Mobile IPv6, (2) a single-homed node using MoVeNet and finally (3) a multi-homed node using MoVeNet. Piggybacking mechanisms are activated. For each run, we simulate 100 scenario repetitions, resulting in up to 800 assessed handovers per run.

6.5.2 *Handover Delay*

Figure 49 shows the cumulative distributions of the handover delay for the three compared mobile nodes. Note that all presented handover delays are from the point of view of the Data Agent.

Analyzing TCP handover delays in graph (a) and (b), MoVeNet reaches extraordinary gains over Mobile IPv6 (4408ms) for handover delay in median by factor 47.41 (93ms) for single-homed mobile nodes and factor 407.45 (11ms) for multi-homed mobile nodes. These significant gains emerge from the MoVeNet retransmission trigger. In about 95% of cases, the TCP retransmission trigger continues more than 1s late. In about 10% of cases, the time loss of the TCP triggering in comparison to MoVeNet is even more than 10s.

The benefit of piggybacking, which avoids the detour via the Control Agent, can be observed by comparison of the UDP results in graphs (c) and (d). Piggybacking is active in the run (d) UDP *MA to CN* and not applicable, i.e. inactive, for the run (c) UDP *CN to MA*. The run (d) shows in median a performance gain of factor 1.78 over Mobile IPv6, attaching signaling information to packets in order to avoid the detour via the Control Agent. In contrast, the performance gain for the UDP run (c) in the opposite direction, where piggybacking is not applicable, is only 1.33 over Mobile IPv6. This improvement is a result of the shorter packet route, which does not take the detour via the CA.

MoVeNet *outperforms Mobile IPv6 regarding handover delay by factor 1.78 with the fast mechanism, respectively 1.33 with its default mechanism.*

Furthermore, there is a significant performance improvement for MoVeNet employed at a multi-homed node in comparison to a single-homed mobile node, reducing median handover delay for MoVeNet nodes in average over the four runs (a-d) by factor 6.7. A single-homed client has to pause transmission at a handover during interface reconfiguration. In contrast, the applied multi-homing-based *make-before-break* paradigm [165, 23, 213] prepares the new network for transmission before handover execution and, in the meanwhile, continues transmission via the previous network. Thus, multi-homing enables an instant change of routing to a second active network without creating any transmission pause.

6.5.3 *Retransmission Trigger*

To stress the MoVeNet retransmission trigger, we create connectivity gaps in the simulated scenario by decreasing the range of the WiFi access networks to about 200meters and use a single-homed MoVeNet Mobile Node. Figure 50 shows the number of transmitted packets over time in two runs with activated, respectively deactivated MoVeNet retransmission trigger. Plateaus in the graph represent transmission pauses, which should be minimized, whereas a rising slope belongs to a running transmission.

Transmission pauses begin for both runs at about the same points in time, meaning that the network gets out of reach for both runs equally. However, we identify a length pattern in transmission pauses before TCP successfully triggered resumption. Exponentially rising timeouts trigger TCP retransmission attempts at $t = 2^n - 1$ seconds after network connection loss. In the example, TCP successfully triggered resumption in three cases after 31 seconds and in the two other cases

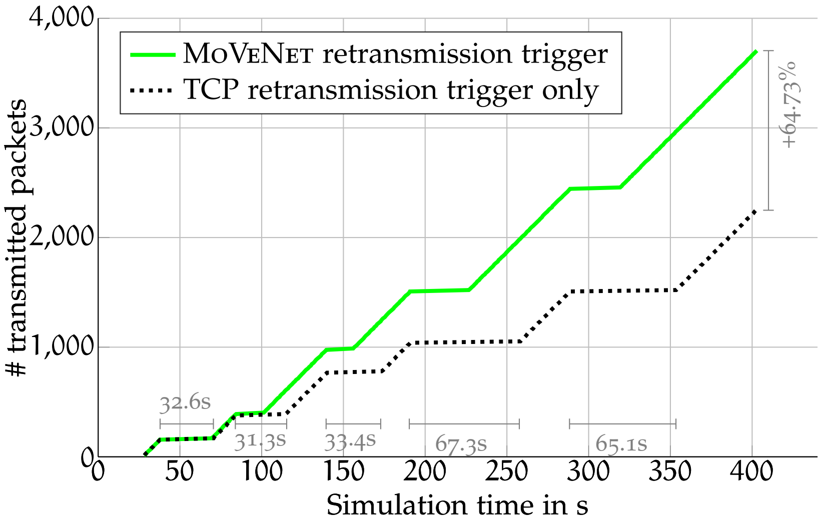


Figure 50: MoVeNet retransmission trigger effect simulation on a single-homed client running a single TCP connection from mobile node to server

Due to MoVeNet's retransmission trigger, 64% additional packets have been transmitted in the scenario. It makes TCP continue transmission as soon as possible.

after 63 seconds. In Section 2.2.4, we analyzed connectivity durations to short- and mid-range networks at different vehicle speeds and identified usual maximum durations of 10 seconds to 1 minute. Accordingly, TCP triggers can easily miss entire networks after a longer connection loss.

In contrast, the green solid line in Figure 50 illustrates the results for the event-triggered MoVeNet enhancement. As soon as the network is ready to transmit data, MoVeNet successfully triggers the TCP connection and makes the transmission continue. In the example, this simple mechanism leads to an improvement of 64.73% additionally transmitted packets, showing the advantages of the introduced triggering approach.

6.6 PROTOTYPICAL EVALUATION

To assess the real-world applicability of MoVeNet, we developed a prototype and investigated the complexity of its networking and processing operations. Moreover, we stress on the performance difference between the default and the piggybacking-based initialization and IP synchronization mechanisms and evaluate system performance.

6.6.1 *Linux Prototype Design*

Three core functions of a mobility management protocol are, firstly, providing a persistent virtual network access for the mobile node connections, avoiding interruptions, secondly, the mapping between a changing real network access to the persistent virtual one at involved entities and, thirdly, synchronization of involved entities to complete the distributed mapping. We target to use available Linux func-

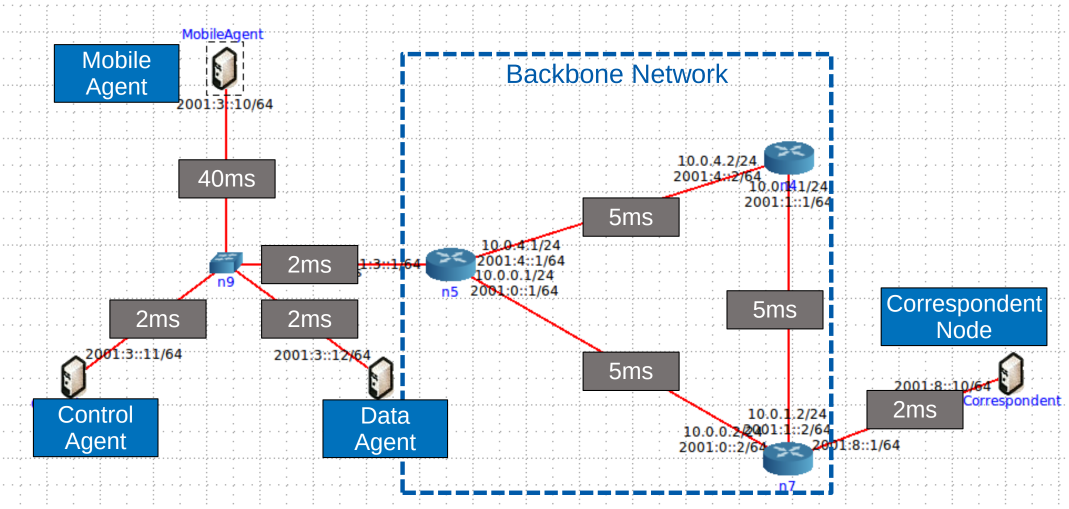


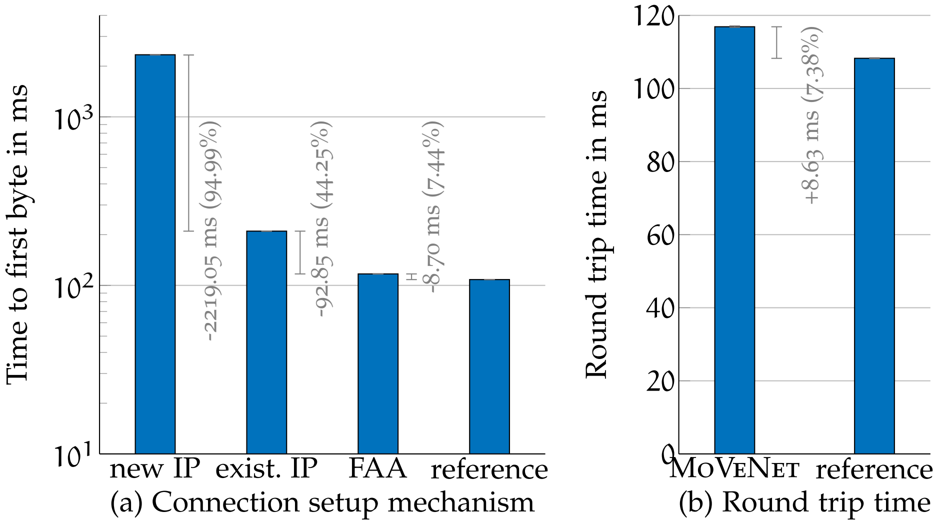
Figure 51: Emulated scenario for MoVeNet protocol evaluation using CORE emulator. Links are annotated with their round trip time.

tions to implement those features through system configuration without changes of the kernel, ensuring simple setup and upgrading of systems.

The Linux prototype introduces a dummy interface as a virtual stable point of access and processes MoVeNet *headers in user space.*

As MoVeNet operates on the network layer, our Linux prototype employs a *dummy network interface* configured with a static IP address representing a persistent virtual network access for all connections. A dummy interface usually accepts but drops all packets except those addressed to localhost. To capture packets and safe them from being dropped, we employ the Linux *netfilter* framework, which usually serves for defining firewall rules. It provides the opportunity to forward packets in the network stack to a queue, accessible from user space. We configure it to employ this option to all packets sent from the persistent node IP address. Packet processing in the user space simplifies implementation and decouples the protocol operation from the Linux kernel, ensuring easy system integration. However, the transfer between spaces introduces an additional processing delay. This implementation approach has been applied for several mobility management protocols of different kinds [87, 157, 18, 158]. The prototype modifies the packets according to the mapping approach of MoVeNet, as detailed in Section 6.3.1. Consecutively, the packet with modified IP addresses for routing is pushed back to the network stack, using the netfilter *pre-routing hook*, ensuring that the packet is routed via the network interface as identified by the inserted packet source IP address.

To address the third core function, synchronization of involved entities, we employ a Linux *raw socket*, which bypasses the dummy interface and netfilter process, reaching independence from the network management of MoVeNet for signaling. This independence from the managed connections is important because signaling unlocks access networks for data connections. If signaling used standard connections managed by MoVeNet and if all active network connections fail, no signaling information could pass anymore to Data Agents to unlock new ones, resulting in a system deadlock. Using a raw socket to handle signaling, bypassing the network filters, solves this issue.



(a) Connection setup mechanism

(b) Round trip time

Figure 52: Connection setup time in time-to first byte for MoVeNet allocating a new IP address at DA, using an existing IP address, using Fast Address Allocation (FAA) and reference without MoVeNet (a). Additional round trip time caused by MoVeNet (b).

6.6.2 *Prototype Evaluation Setup*

To create a controlled evaluation environment, we use the Common Open Research Emulator (CORE) [3], which enables the creation of multiple fully functional Linux instances on a single machine interconnected via simulated network links. In contrast to alternatives as Mininet [124], CORE supports simulated IPv6 networks, as required for MoVeNet.

We set up an emulation environment, consisting of a correspondent node with a conventional network stack, a Control Agent, a Data Agent and a mobile node, configured with a MoVeNet Mobile Agent, as shown in in Figure 51. Links connecting these instances have a line delay of 2 ms. The mobile node is connected using a link with 40ms line delay, representing the wireless transmission. In addition, we connect the correspondent node through a backbone network whose switches are interconnected with links with a 5 ms line delay.

6.6.3 *Connection Setup and Round Trip Time*

FAA speeds up the connection initialization by 44.25% compared to the default mechanism.

Setting up the handover-enabled data bridge between Mobile Agent and a Data Agent for a new connection requires some time, differing for the two complementary initialization mechanisms, targeting either a low route delay or low setup latency. To assess the different connection setup mechanisms, we evaluate the resulting time to first byte of a UDP transmission to the Data Agent, covering MoVeNet connection setup measured as the time till the first byte arrives at the correspondent node and the round trip time. Firstly, we address the issue of creating a new IP address at the Data Agent, which has to be announced in the surrounding network through lower layer routing protocols before its use. This case is denoted in

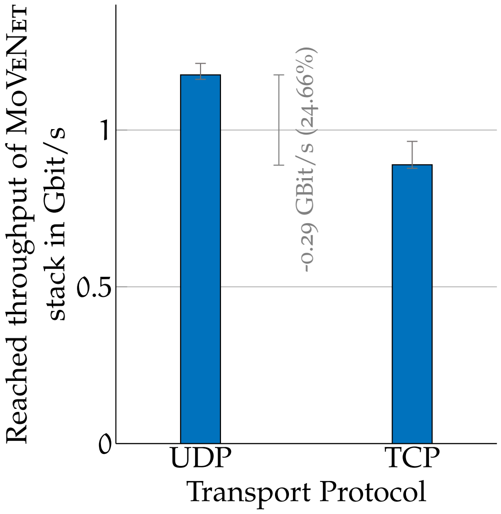


Figure 53: UDP and TCP throughput with MoVeNet using an ideal network link

Figure 52 (a) with the most left bar, new IP, employing MoVeNet's default mechanism, using the Control Agent as broker for Data Agent selection and initialization, and generating a new Data Agent IP address for the mobile node. The the first data can be sent in median after 2335.99 ms, mainly caused by the setup of a novel IP address at the Data Agent. If the same mechanism is used, but the Data Agent reuses an already existing IP address, the delay requires in median only 209.81 ms in the evaluation scenario, which is 94.99% less. To avoid these high latencies, we firmly recommend Data Agents to allocate a bunch of IP addresses for the following operation during their initialization and whenever they are close to running out of unused addresses.

In comparison to the default mechanism, the piggybacking-based fast address allocation (FAA) connection initialization mechanism reduces the connection setup delay by 92.85 ms (44.25%), reaching a median delay of 116.96 ms, in expense for a restricted Data Agent selection and some additional per-packet overhead. The reduced delay of the 92.85 ms is dominated by the time required in the default mechanism for transferring the signaling packet, passing the wireless link for the Data Agent Initialization twice, for request and response (in total about 80 ms). Compared to a reference connection setup without using MoVeNet, FAA creates only an additional delay of 8.70 ms (7.44%) in the evaluation scenario.

The round trip time is increased from the detour via the DA, adding 8.63ms additional

The Data Agent as proxy introduces a routing detour, causing an additional route delay. In addition, the packet processing at the four stages at Data and Mobile Agent requires a certain time. As illustrated in Figure 52 (b), these two delay sources lead in the evaluation scenario to a round trip time delay increase by 8.63 ms (7.38%), differing only insignificantly from that for packet forwarding and connection setup using FAA. The similarity originates from using the same detour for sending, and producing a slightly higher processing effort for initialization than for forwarding. Using well-selected Data Agents as connection proxy servers, as discussed in Section 6.3.2, this additional delay can be minimized.

Table 14: MoVeNet prototype component-wise processing time and memory consumption

Description	CPU time	Percentage	Memory allocation
MoVeNet prototype stack	63523.1 ns		1584 Bytes
Sum MoVeNet user space	695.0 ns	1.09% [a]	32 Bytes
Checksum calculation[1]	537.0 ns	77.27% [b]	0 Bytes
Packet serialization[2]	84.7 ns	12.19% [b]	32 Bytes
Packet decoding[3]	21.6 ns	3.11% [b]	0 Bytes
Flow identification[3]	13.7 ns	1.97% [b]	0 Bytes
Address translation[3]	14.0 ns	2.01% [b]	0 Bytes
others[3]	24.0 ns	3.45% [b]	0 Bytes
Sum MoVeNet processing only[3]	73.3 ns	2.44% [c]	0 Bytes
Conventional stack (reference)	3000.5 ns		48 Bytes

[1] may be offloaded to the network interface controller (hardware processing)
[2] due to user space implementation
[3] mandatory processing components of MoVeNet
[a] percentage of MoVeNet prototype stack CPU time
[b] percentage of Sum MoVeNet user space CPU time
[c] percentage of default stack CPU time

6.6.4 *Throughput Impact and Scalability*

To assess the applicability of the prototype, we evaluate the maximum throughput which the prototype implementation can deliver on our test system. Therefore, we send TCP and UDP data from the mobile node to the correspondent node using the Linux tool *iperf* in the emulated setup of Figure 51. The simulated network links and intermediate nodes provide unlimited throughput. The setup reaches a throughput of 1.17 Gbits/s for UDP and 0.89 Gbits/s TCP throughput, corresponding to 24.66% less, as shown in Figure 53. This high performance ascertains the applicability of the implemented MoVeNet prototype for further protocol evaluation. The limitation seems to be caused by delays in the packet processing pipeline, introduced from the user space transfer.

To analyze the packet processing delays introduced by the MoVeNet prototype, we present component-wise latency benchmarks in Table 14, ascribed with the delay sources. Therefore, we send a packet from an application, transfer it to the user space with netfilter, process it in MoVeNet and finally drop it. The packet requires in average 63523.1 ns to pass the complete prototype stack. However, packet-processing of MoVeNet in the user space takes in average only 695.0 ns per packet, causing just 1.09% of the stack's delay. Analyzing the CPU time required for processing the components of the MoVeNet user space implementation further, we discover that 77.27% (537.0 ns) are caused from packet checksum calculation, which can be processed in hardware at the network interface controller in an optimized kernel implementation. In addition, 12.19% (84.7 ns) belong to packet serialization, which is also required in the user space implementation only. Mandatory components for a kernel implementation, dominated by including packet decoding, flow identification, address translation consume an accumulated CPU time of 73.3 ns only. In relation to sending a packet using the conventional network stack, requiring in average 3000.5 ns on our evaluation system, this increases the packet processing delay by only 2.44%. This analysis shows that MoVeNet packet processing adds only an insignificant delay in packet processing, qualifying the concept for real world application.

Mandatory parts of MoVeNet *processing would, in an optimized implementation, increase the processing time of the conventional stack by approximately 2.44%.*

6.7 transmission planning as control entity for movenet

MoVeNet provides the means to maintain valid network routes for mobile nodes and use them to dispatch packets. In contrast to other mobility management protocols, MoVeNet does not control the mapping of data flows to these routes itself. In contrast, this strategic decision is addressed by our transmission planning and adaptation, as presented in the previous chapters. These sophisticated algorithms replace simplistic methods for network selection and control the actual distribution of data.

The decision of how to integrate these planning algorithms into the MoVeNet architecture depends on the identified prediction error impact, as analyzed in Chapter 5. The adaptation can effectively mitigate the impact of moderate prediction errors from movement and network characteristic changes. In contrast, data flow prediction errors have a strong and hardly addressable impact on transmission

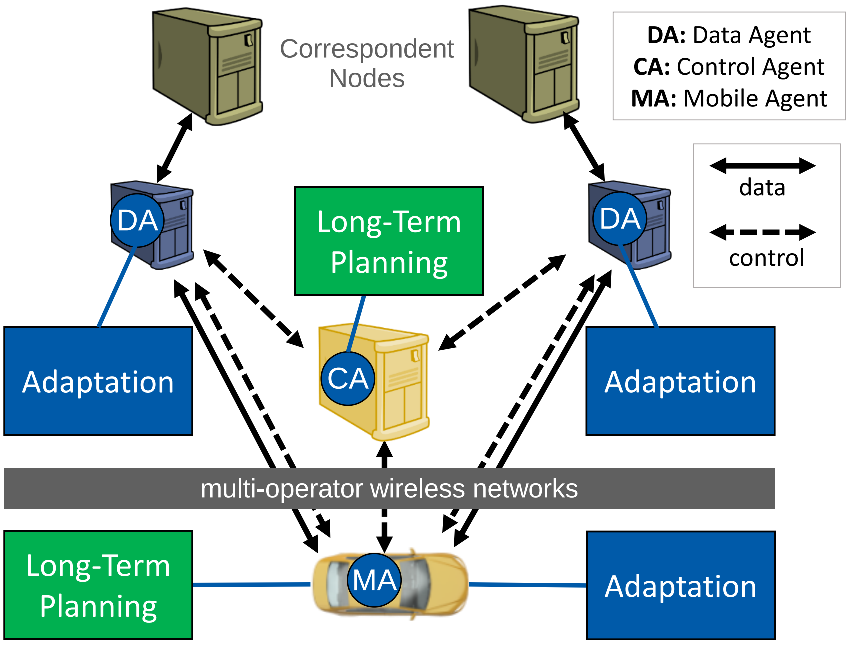


Figure 54: Intelligent Connectivity Management System Architecture

performance. As data flow changes are usually triggered by user interaction, the mobile node is predestined for initial calculation of long-term plans, using Joint Transmission Planning, as presented in Chapter 4. As for IP synchronization, the Control Agent can be used as a broker for transmission plan synchronization between the agents.

Adaptation must be applied to outgoing and incoming data for the mobile node and, hence, has to be located at the Mobile Agent and the Data Agents, as illustrated in Figure 54. An open challenge remains to the distributed adaptation at those entities, as it is time-critical to sustain the benefits from planning. For outgoing data from the mobile node, adaptation can be applied straight because the node monitors all environmental changes and, thus, has essential required information. However, incoming data traffic to the Mobile Agent from Data Agents faces two challenges. Firstly, Data Agents either have to receive updates on environmental changes or induce them from the sent and received data traffic. Secondly, data is routed via different Data Agents, which requires a low latency coordination of their adaptation, each observing and managing only a subset of the active data flows. The distributed system faces a trade-off between full information sharing on the one hand, which produces significant management overhead, partially even via the wireless links, and may introduce extra delays for adaptation. On the other hand, not sharing all information may result in an inconsistency or divergence of adapted plans at the different entities, with which the system has to cope. We expect a dependency from the number of employed Data Agents on this potential inconsistency. Furthermore, the long-term re-planning rate may affect these dynamics as well, resetting the inconsistency to zero. The resulting system dynamics require deeper analysis, which is beyond the scope of this dissertation, and open new research challenges for the future.

6.8 summary and conclusions

Mobility management for Vehicular Networking (MoVeNet) is a user-controlled and multi-homing-based distributed mobility management approach, creating cross-operator network resource pools. Available networks can be employed dynamically for selective distribution of the user's data traffic to available links, enabling the execution of data transmission plans. Its distributed architecture, composed of the Mobile Agent (MA) located at the mobile node, Data Agents (DA) for data forwarding and a Control Agent (CA) that supports the Mobile Agent in system management, provides system robustness, scalability and compatibility to backend nodes with a conventional network stack.

Novel complementary control mechanisms of MoVeNet for connection initialization and route announcement enable demand-based and connection-specific optimization of non-functional requirements. MoVeNet employs a publish-subscribe-based mechanism to distribute route announcements to all employed Data Agents to minimize the overhead. For latency-sensitive connections, the complementary mechanism may send data instantly via a new route without prior announcement, piggybacking required signaling information, which eliminates handover latency. Furthermore, the default initialization mechanism covers an optimized Data Agent selection for connection-specific route latency reduction, selected by the Control Agent. Its complementary mechanism, *Fast Address Allocation* (FAA), contacts a known Data Agent instantly, sending signaling information piggybacked with the data to eliminate connection setup latency.

MoVeNet introduces a novel IP mapping approach at Data Agents (DA), namely *IPv6 address multiplexing*, which reduces per-packet overhead and eliminates transport layer dependence for packet forwarding. Using IPv6, nodes receive an address space from which they can select an own address for their interface. Our approach targets using multiple Data Agent IP addresses at the same time, one for each managed connection. Accordingly, the Mobile Agent sends packets of each connection to a dedicated DA IP address from a reserved range. The DA and MA can identify the connection including the packet's forwarding destination from the DA IP address, which the packet carries. As current Linux systems show degrading performance with more than about 1000 employed IP addresses, we propose a hybrid approach, using dedicated IP addresses per MA, which reduces the number of required IP addresses significantly. To identify the correspondent node for packet forwarding, MoVeNet attaches a correspondent node ID (CN ID) to each packet between MA and DA, which creates only 4 bytes overhead.

Furthermore, we show by simulation that TCP's exponentially rising retransmission probing is inefficient in environments with sparse network coverage and fast moving clients, e.g. a connected vehicle. To address this problem, MoVeNet introduces an external event-based trigger to make TCP continue transmission as soon as possible. Therefore, the Mobile Agent stores always the last sent and the last received packet. As soon as MoVeNet recognizes an available new route, the trigger re-sends the stored packets, triggering TCP to continue data transmission.

We evaluated MoVeNet using simulation and a Linux-based prototype. We especially highlight the following three main contributions, ordered by their importance.

1. Complementary mechanisms provide demand-based and connection-specific performance optimizations on management processes and create robustness for the distributed system.

   Measured with our Linux prototype system in an evaluation setup, the low-latency initialization mechanism FAA reduces the initialization delay by 44.25% compared to the default mechanism. The resulting delay covers only 7.44% (8.70ms) additional time to first byte relative to a reference connection initialization without MoVeNet. Furthermore, the fast piggybacking-based route announcement mechanism sends signaling along with data, eliminating additional announcement delays completely. Moreover, in the case of node failure of either a Data Agent or the Control Agent, one of the complementary mechanisms is still applicable, which ensure seamless protocol operation.

2. The novel IP mapping approach of MoVeNet simplifies packet forwarding at the proxy, the Data Agent. It employs dedicated Data Agent IP addresses per connection, respectively per Mobile Agent in the hybrid approach, for identification of the packet's destination. The new approach has three advantages. Firstly, it simplifies the processing for forwarding, relying only on the IP header, which compared to network address translation creates transport layer independence. Secondly, it eliminates per-packet overhead, respectively reduces it to 4 bytes, compared to 40 bytes for tunneling. Thirdly, the hybrid approach introduces return routability, i.e. the correspondent node is able to send data to the mobile node via the managed connection despite changing IP addresses, as long as the DA IP address is active for usage with the CN.

3. The event-based retransmission trigger of MoVeNet addresses TCP's inefficient connection recovery in the connected vehicle scenario with sparse network availability. It increases the number of transmitted packets in a simulated scenario by 64.73%, demonstrating the effectiveness of this mechanism.

MoVeNet resolves the mobility management restriction of many approaches to resources of a single network operator, while providing substantially improved management and routing characteristics. Together with the contributions of transmission planning and adaptation, MoVeNet pushes strategic joint time-network selection towards real systems in order to exploit the hidden optimization potentials discovered in this dissertation.

7 CONCLUSIONS AND OUTLOOK

Future autonomous driving systems release drivers from their task to control the vehicle, letting them use their travel time productively or for entertainment. Resulting future peak demands in network resources during rush hours cannot be satisfied from mobile networks, even considering expected network improvements [58]. Hence, smarter ways to cope with this problem are required, defining the goal of this dissertation. In the following Section 7.1, we summarize our approach of using strategic data transmission with joint time-network selection and highlight our contributions. Consecutively, we give an outlook on future works, stating new research questions based on our contributions.

7.1 CONCLUSIONS

This dissertation addresses the improvement of the perceived transmission quality for connected vehicle users in order to treat future expected connectivity demands that mobile networks alone will not be able to satisfy [58]. To achieve this goal, we designed data transmission strategies and mechanisms that exploit the full potential of heterogeneous wireless network environments. To this end, we highlight our four main contributions in the following.

To assess the perceived quality of transmissions, we presented a novel transmission rating model, constituting our first main contribution. The rating model focuses on application QoS *satisfaction* traded-off by monetary cost, addressing the user's perceived quality of transmissions. We combined components from transmission time selection models and network selection models. However, these rating approaches are incompatible with respect to their throughput requirement models. While network selection employs a fixed parameter to model the throughput requirement, transmission time selection defines deadlines by which transmission of data sets should be completed. Hence, they address either continuous or delay-tolerant transmission. To solve this conflict, we generalize existing approaches in our novel throughput requirement model in characterizing throughput by its fundamental definition: an amount of data transmitted over time. By parametrizing the time window length of the requirement, throughput requirements can range from continuous and instant transfer for a time window length of one time slot to broad temporal flexibility for large windows. Hence, the new model is able to treat transmissions with different delay-tolerance characteristics, which have been modeled independently before. Connection-specific weights for each requirement, complete the generalization of state-of-the-art models and allow rating of transmissions in a unified way.

Our second main contribution is the development of a novel time-network selection strategy, realized in our Joint Transmission Planning (JTP) approach. It creates transmission plans that avoid application QoS requirement violations and reduce monetary cost due to moving transmissions to points in time that provide the best-matching transmission opportunities within a certain time horizon.

To evaluate the performance of the joint time-network selection strategy, we compare its performance to that of leading strategies from the state-of-the-art. JTP outperforms the state-of-the-art approaches in nearly all cases by 7-26%. Evaluation reveals that pure network selection suffers in environments with sparse network resources, offering few choice and significant changes in network quality over time, as they consider only the current situation. In contrast, pure transmission time selection approaches suffer in environments with many different networks due to the lack of a sophisticated network selection. In extensive scenario variation, our presented approach JTP is the only one that shows over a planning time horizon overall performance robustness reaching a constantly high performance of 87-91% compared to the optimum. These results demonstrate the superiority of our joint time-network selection approach.

As JTP is a planning-based approach, strict execution of its plans lacks robustness against prediction errors, i.e. due to incomplete information or environmental changes. To sustain the benefits of joint time-network selection in practice, we designed a transmission plan adaptation approach as our third main contribution. During plan execution, it modifies the current transmission in order react to environmental changes. Therefore, our adaptation approach employs an opportunistic transmission strategy as underlying algorithm and introduces constraints forcing it to follow a transmission plan. As a reaction to environmental changes, the adaptation approach selectively relaxes the introduced constraints to enable partial opportunistic transmission instead of following the plan strictly. In particular, we designed three mechanisms, each handling one kind of environmental change, namely movement, network and data flow changes. Based on the prediction error strength, the mechanisms introduce dynamic offsets to modify the plan and relax the constraints. Due to the introduced mechanisms and even under strong movement, respectively network changes, the adaptation sustains 57%, respectively 36% of the performance gain from planning. For moderate data flow changes up to 61% can be sustained. The performance of the adaptation converges to the performance of the underlying opportunistic approach for growing changes, providing a robust lower bound. Conclusively, adaptation effectively copes with prediction errors caused by incomplete information or environmental changes, sustaining a significant share of the performance gain from planning.

To push transmission planning towards real systems, we developed the protocol *Mobility management for Vehicular Networking* (MoVeNet) as our fourth contribution, which pools available network resources and enables flexible packet dispatching without causing connection interruptions. Existing protocols suffer from individual weaknesses, which impede their use in the connected vehicle scenario. Excluding certain IP data traffic, restrictions to networks within a limited managed domain, or inefficiency due to long connection setup and handover delays are just a few examples that disqualify existing approaches.

MoVeNet introduces a distributed architecture, composed of Data Agents, a Control Agent and the Mobile Agent. The Mobile Agent (MA), located at the mobile node, creates a stable point of access at the network layer, hiding flexible data routing and multi-homing from upper layer protocols. Data Agents (DA) serve as end-points for this flexible routing, hiding MoVeNet protocol operation from the mobile node's communication partners to provide compatibility. Furthermore, the Control Agent (CA) supports the MA in management tasks and DA orchestration, reducing the overhead via the wireless links.

MoVeNet employs complementary mechanisms for connection initialization and for route announcement, enabling handovers. For each, a first mechanism employs the CA and a complementary mechanism contacts DAs directly to reduce latency. The mechanisms can be selected individually to address specific QoS requirements for each connection. The first route announcement mechanism targets a low overhead by employing the CA as publish-subscribe broker for control information. Its complementary mechanism addresses a low-latency handover by eliminating delays from routing detours. For connection initialization, the first mechanism selects and initializes a new DA based on its location to minimize the connection-specific route latency. The second mechanism skips this initialization and employs an existing DA, attaching required signaling information to the data packets, which are instantly sent to the DA. This low-latency mechanism eliminates additional setup delays and, in our tests, reduced the time-to-first-byte significantly by 44%. Furthermore, the proposed mechanisms introduce system robustness, allowing MoVeNet to cope with failures of the CA or DAs and to continue operation seamlessly. Beyond that, MoVeNet's novel event-based packet retransmission trigger mechanism reduces transmission recovery delays of TCP in sparse network environments, achieving in our evaluation up to 64% of additionally transmitted packets. Finally, MoVeNet implements a new IP mapping system for packet forwarding at DAs, achieving independence from the transport layer and reducing the per-packet overhead to 0 to 4 bytes. Using the new IP mapping approach, the DA creates a dedicated IP address for each mobile node, which is employed for all its communication. Hence, for packet forwarding at the DA, the desired destination of all incoming packets can be identified efficiently from the IP header. Evaluation of MoVeNet demonstrates its feasibility and efficiency for the connected vehicle scenario.

In summary, this dissertation provides a solution to mitigate the impact of expected future resource bottlenecks of mobile networks by employing strategic data transmission, exploiting the full potential of heterogeneous wireless network environments. The collaboration of the presented mechanisms leads to a system in which flexible route selection in heterogeneous networks is enabled by our optimized mobility management approach MoVeNet. To control its transmissions and to cope with rapid environmental changes and incomplete information, our solution employs the presented adaptation algorithm. It modifies data transmission plans from the key component of this work, our joint time-network selecting transmission strategy Joint Transmission Planning, which fosters application QoS satisfaction traded off by monetary cost. This way, the demonstrated significant improvement of the perceived transmission quality provided by strategic data transmission can be exploited in future systems.

7.2 OUTLOOK

Insights gained in this dissertation open new perspectives and challenges for the future role of time-network selection in network management. In the following, we highlight future research directions referring to the single-client system and the access network ecosystem.

- Final integration of the presented mechanisms into a holistic system remains as an open task, enabling real-world application. While MoVeNet exists as a working protocol prototype, it still lacks a full integration of JTP, the adaptation and prediction approaches as generic containers. Especially the selection and integration of network and data flow prediction offers further research opportunities. We propose to use connectivity maps for network predictions, extended by short-term network estimation, inspired work of Bui et al. [27] and Poegel et al. [166]. For data flow prediction, historical patterns show good results [15], which might be enhanced further with pre-fetching models [116].

- The coordination of responsive transmission plan adaptation over multiple entities introduces introduces new challenges. To coordinate a distributed adaptation, Data Agents may exchange information. This exchange produces a considerable overhead and, moreover, may introduce a latency to reactions. Accordingly, for these distributed algorithms, an extensive information exchange for virtually deterministic cooperation has to be traded-off by working on incomplete or aggregated information with a context-dependent information granularity. This trade-off is certainly influenced by the transmission plan updating frequency because updates sent to all entities eliminate the information gap at distributed entities. To reduce the demand for communication, only the next few time slots of a plan should be distributed, preferably using the Control Agent as a publish-subscribe broker, inspired from our efficient handover signaling mechanism.

- From a network operator's point of view, transmission plans offer valuable information about the future resource demand of wireless access networks in space-time dimension. It helps them to prepare their networks proactively for the prospective demand without an expensive over-provisioning of resources in the access and backhaul networks. Especially in future 5G networks with very flexible resource allocation capabilities, this may be profitable, as unused resources may even be traded between different operators or additional resources can flexibly be acquired to satisfy the expected demand. Furthermore, operators may offer dynamic pricing strategies for ensuring QoS requirement satisfaction with a higher probability or as an incentive to use networks when allocated resources are idle.

Our contributions on strategic data transmission using heterogeneous wireless networks constitute the foundation for further research in the above-mentioned directions.

BIBLIOGRAPHY

[1] Osama Abboud. *Quality Adaptation in P2P Video Streaming - Enhancing Performance and Supporting Heterogeneity using Scalable Video Coding*. PhD thesis, Technische Universität Darmstadt, 2012.

[2] Atiq Ahmed, Merghem Boulahia, and Dominique Ga. Enabling Vertical Handover Decisions in Heterogeneous Wireless Networks : State-of-the-Art and Classification. *IEEE Communications Surveys & Tutorials*, 16(2):776–811, 2014.

[3] Jeff Ahrenholz, Claudiu Danilov, Thomas R. Henderson, and Jae H. Kim. CORE: A Real-Time Network Emulator. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2008.

[4] Alaa Alhamoud, Vaidehi Muradi, Doreen Böhnstedt, and Ralf Steinmetz. Activity Recognition in Multi-User Environments Using Techniques of Multi-label Classification. In *Proceedings of the ACM International Conference on the Internet of Things (IoT)*, 2016.

[5] Ali F. Almutairi, Mohamed A. Landolsi, and Aliaa O. Al-Hawaj. Weighting Selection in GRA-based MADM for Vertical Handover in Wireless Networks. In *Proceedings of the IEE International Conference on Computer Modelling and Simulation (UKSim)*, 2016.

[6] Halis Altug. *Design of a Mobility Protocol for Vehicular Networking*. Master thesis, Technische Universität Darmstadt, 2015.

[7] Karl Andersson, Daniel Granlund, Muslim Elkotob, and Christer Åhlund. Bandwidth Efficient Mobility Management for Heterogeneous Wireless Networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*, 2010.

[8] Sergey Andreev, Mikhail Gerasimenko, Olga Galinina, Yevgeni Koucheryavy, Nageen Himayat, Shu-Ping Ping Yeh, and Shilpa Talwar. Intelligent Access Network Selection in Converged Multi-Radio Heterogeneous Networks. *IEEE Wireless Communications*, 21(6):86–96, 2014.

[9] Jeffrey Andrews, Sarabjot Singh, Qiaoyang Ye, Xingqin Lin, and Harpreet Dhillon. An Overview of Load Balancing in HetNets: Old Myths and Open Problems. *IEEE Wireless Communications*, 21(2):18–25, apr 2014.

[10] Zoe Antoniou and Ioannis Stavrakakis. An Efficient, Deadline Credit-based Transport Scheme for Prerecorded, Semi-Soft Continuous Media Applicatins. *IEEE/ACM Transactions on Networking*, 10(5), 2002.

[11] J. Arkko, C. Vogt, and W. Haddad. Enhanced Route Optimization for Mobile IPv6 (RFC 4866). 2007.

[12] T. Aura. Cryptographically Generated Address (CGA) (RFC 3972). 2005.

[13] Alaa Awad, Amr Mohamed, and Carla-Fabiana Chiasserini. Dynamic Network Selection in Heterogeneous Wireless Networks: A User-Centric Scheme for Improved Delivery. *IEEE Consumer Electronics Magazine*, 6(1):53–60, 2017.

[14] I. Aydin, W. Seok, and C.-C. Shen. Cellular SCTP: a transport-layer approach to Internet mobility. In *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2003.

[15] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting Mobile 3G Using WiFi. *Proceedings ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2010.

[16] Farooq Bari and Victor C M Leung. Automated Network Selection in a Heterogeneous Wireless Network Environment. *IEEE Network*, 21(1), 2007.

[17] Thomas Jr Barnett, Arielle Sumits, Shruti Jain, and Usha Andra. Cisco Visual Networking Index (VNI) Update Global Mobile Data Traffic Forecast. *Vni*, 2017.

[18] Sébastien Barré. LinShim6 Documentation - Implementation of the Shim6 protocol. Technical report, 2008.

[19] Jan-Thomas Becker. *Optimization and Prototype Development of the Mobility Management Protocol MoVeNet*. Bachelor thesis, Technische Universität Darmstadt, 2017.

[20] Nico Becker, Amr Rizk, and Markus Fidler. A Measurement Study on the Application-Level Performance of LTE. In *Proceedings of the IEEE IFIP Networking Conference*, 2014.

[21] Y. Bejerano and Seung-Jae Han. Cell Breathing Techniques for Load Balancing in Wireless LANs. *IEEE Transactions on Mobile Computing*, 8(6):735–749, jun 2009.

[22] Annette Bieniusa, Marek Zawirski, Nuno Preguiça, Marc Shapiro, Carlos Baquero, Valter Balegas, and Sérgio Duarte. An Optimized Conflict-free Replicated Set. Technical report, 2012.

[23] Roland Bless, Joachim Hillebrand, Christian Prehofer, and Martina Zitterbart. A Quality-of-Service Signaling Architecture for Seamless Handover Support in Next Generation, IP-Based Mobile Networks. *Wireless Personal Communications*, 43(3):817–835, 2007.

[24] BMW AG. BMW ConnectedDrive Services. Accessed at 2017-02-11. URL http://www.bmw.de/de/topics/faszination-bmw/connecteddrive/digital-services/connecteddrive-services.html.

[25] George E. P. Box, J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley, 2005.

[26] Nicola Bui and Joerg Widmer. Mobile Network Resource Optimization under Imperfect Prediction. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.

[27] Nicola Bui, Matteo Cesana, S. Amir Hosseini, Qi Liao, Ilaria Malanchini, and Joerg Widmer. A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques. *IEEE Communications Surveys & Tutorials*, 2017.

[28] Alan Bundy and Lincoln Wallen. Breadth-First Search. In *Catalogue of Artificial Intelligence Tools*. Springer Berlin Heidelberg, 1984.

[29] D. Burgstahler, M. Pelzer, A. Lotz, F. Knapp, H. Pu, T. Rueckelt, and R. Steinmetz. A concept for a C2X-based crossroad assistant. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2015*, 2015. ISBN 9781479984251. doi: 10.1109/PERCOMW.2015.7134063.

[30] Daniel Burgstahler, Nils Richerzhagen, Frank Englert, Ronny Hans, and Ralf Steinmetz. Switching Push and Pull: An Energy Efficient Notification Approach. *IEEE Internetaional Conference on Mobile Services*, 2014.

[31] Daniel Burgstahler, Tobias Meuser, Ulrich Lampe, Doreen Boehnstedt, and Ralf Steinmetz. ProbSense.KOM: A Probabilistic Sensing Approach for Gathering Vehicular Sensed Data. In *Proceedings of the IEEE International Conference on Mobile Services (MS)*, 2016.

[32] Daniel Burgstahler, Tobias Meuser, Ulrich Lampe, Doreen Boehnstedt, and Ralf Steinmetz. ProbSense.KOM: A Probabilistic Sensing Approach for Gathering Vehicular Sensed Data. In *Proceedings of the IEEE International Conference on Mobile Services (MS)*. IEEE, 2016.

[33] Daniel Burgstahler, Christhoph Peusens, Doreen Böhnstedt, and Ralf Steinmetz. Horizon. KOM: A First Step Towards an Open Vehicular Horizon Provider. In *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 2016.

[34] Daniel Burgstahler, Athiona Xhoga, Christoph Peusens, Martin Möbus, Doreen Böhnstedt, and Ralf Steinmetz. RemoteHorizon.KOM: Dynamic Cloud-based eHorizon. In *Proceedings of the Automotive meets Electronics (AmE)*, 2016.

[35] Daniel Burgstahler, Martin Möbus, Tobias Meuser, Doreen Böhnstedt, and Ralf Steinmetz. A Categorization Scheme for Information Demands of Future Connected ADAS. *Proceedings of the Automotive meets Electronics (AmE)*, 2017.

[36] Carsten Büttner and Tobias Rückelt. Sicheres Hochladen , Austauschen und Verteilen von Daten in einem Car2X Systemverbund. In *Proceedings of the Automotive meets Electronics (AmE)*, 2016.

[37] Ian Catling and Frans Op de Beek. SOCRATES: System of Cellular Radio for Traffic Efficiency and Safety. In *Proceedings of the IEEE Vehicle Navigation and Information Systems Conference*. IEEE, 1991.

[38] Sandra Céspedes and Xuemin Sherman Shen. IP Mobility Management for Vehicular Communication Networks : Challenges and Solutions. *IEEE Communications Magazine*, 49(5), 2011.

[39] B.R. Chandavarkar and Ram Mohana Reddy Guddeti. Simplified and Improved Multiple Attributes Alternate Ranking Method for Vertical Handover Decision in Heterogeneous Wireless Networks. *Elsevier Computer Communications*, 83:81–97, 2016.

[40] Nan Cheng, Ning Lu, Ning Zhang, Nan Cheng, Ning Lu, Ning Zhang, Xuemin Sherman Shen, and Jon W Mark. Vehicular WiFi Offloading: Challenges and Solutions. *Elsevier Vehicular Communications*, 13(21), 2014.

[41] Man Hon Cheung and Jianwei Huang. DAWN: Delay-Aware Wi-Fi Offloading and Network Selection. *IEEE Journal on Selected Areas in Communications*, 33(6):1214–1223, 2015.

[42] Edwin Kah Pin Chong and Stanislaw Zak. *An Introduction to Optimization*. Wiley Interscience, third edition, 2013.

[43] Chung-Ming Huang and Chao-Hsien Lee. Signal Reduction and Local Route Optimization of SIP-Based Network Mobility. In *IEEE Symposium on Computers and Communications (ISCC'06)*, 2006. ISBN 0-7695-2588-1. doi: 10.1109/ISCC.2006.146.

[44] R. Coltun, D. Ferguson, J. Moy, and A. Lindem. Open Shortest Path First Version for IPv6 (OSPF for IPv6) (RFC 5340). 2008.

[45] CONVERGE Consortium. CONVERGE Deliverable D1.1. Accessed at 2014-02-11, 2013. URL http://converge-online.de/doc/download/Deliverable-WP1-D11{_}01{_}01{_}00.pdf.

[46] D.J. Dailey. Travel-Time Estimation Using Cross-Correlation Techniques. *Elsevier Journal on Transportation Research Part B: Methodological*, 27(2), 1993.

[47] Daimler AG. Mercedes me: Connecten Sie sich mit Ihrem Auto. Accessed at 2017-02-11. URL https://www.mercedes-benz.com/de/mercedes-me/konnektivitaet/.

[48] V. Davarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol (RFC 3963). 2005.

[49] Antonio de la Olivia, Albert Banches, and Ignacio Soto. An Overview of IEEE 802.21: Media-Independent Handover Services. *IEEE Wireless Communications*, 15(4), 2008.

[50] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification (RFC 2460). 1998.

[51] Andrei Deftu and Jan Griebsch. A Scalable Conflict-Free Replicated Set Data Type. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2013.

[52] Deutsche Bahn AG. WLAN im ICE. Accessed at 2017-02-11. URL https://www.bahn.de/p/view/service/zug/railnet{_}ice{_}bahnhof.shtml.

[53] Amine Dhraief and Nicolas Montavont. Towards Mobility And Multihoming Unification The SHIM6 Protocol. In *Proceedings of the IEEE IEEE Wireless Communications and Networking Conference (WCNC)*, 2008.

[54] DriveC2X Consortium. Connecting vehicles for Save, Comfortable and Green Driving on European Roads. Accessed on 2017-05-30, 2011. URL http://www.drive-c2x.eu/project.

[55] Zhiyong Du, Qihui Wu, and Panlong Yang. User-Demand-Aware Wireless Network Selection : A Localized Cooperation Approach. *IEEE Transactions on Vehicular Technology*, 63(9):4492–4507, 2014.

[56] Zhiyong Du, Qihui Wu, and Panlong Yang. Dynamic User Demand Driven Online Network Selection. *IEEE Communications Letters*, 18(3):419–422, 2014.

[57] W.M. Eddy. At What Layer Does Mobility Belong? *IEEE Communications Magazine*, 42(10), 2004.

[58] Edward H. Baker, David Crusius, Marco Fischer, Walter Gerling, Kaushik Gnanaserakan, Henning Kerstan, Felix Kuhnert, Julia Kusber, Joachim Mohs, Manuel Schulte, Jonas Seyfferth, Juliane Stephan, and Trent Warnke. Connected Car Report 2016: Opportunities, Risk, and Turmoil on the Road to Autonomous Vehicles. Technical report, Strategy&, 2016.

[59] Melhem Ei Helou, Samer Lahoudt, Marc Ibrahim, and Kinda Khawam. Satisfaction-based Radio Access Technology Selection in Heterogeneous Wireless Networks. In *Proceedings of the IFIP Wireless Days (WD)*, 2013.

[60] Levent Ekiz, Christian Lotterrnann, David Ohmann, and Thang Trant. Potential of Cooperative Information for Vertical Handover Decision Algorithms. In *Proceedings of the International IEEE Annual Conference on Intelligent Transportation Systems (ITSC)*, 2013. ISBN 9781479929146.

[61] S.R. Ely. Development of the European Radio-Data-System RDS for VHF/FM Broadcasting. In *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, 1986.

[62] Ericsson. Mobility Report. Technical report, 2016.

[63] Petro P. Ernest, Olabisi E. Falowo, and H. Anthony Chan. Network-Based Distributed Mobility Management: Design and Analysis. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013.

[64] ETSI. TR 102 638 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions. Technical report, 2009.

[65] ETSI. TS 102 637-1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 1 : Functional Requirements. Technical report, 2010.

[66] ETSI. TS 102 636-4-1 - v1.1.1 - Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Subpart 1: Media-Independent Functionality, 2011.

[67] ETSI. TS 102 636-6-1 - v1.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols. 2011.

[68] ETSI. EN 302 663 - V1.2.0 - Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. Technical report, 2012.

[69] ETSI. TS 124 312 v13.3.0: Access Network Discovery and Selection Function (ANDSF). Technical report, 2016.

[70] ETSI. TS 45.001: 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; GSM/EDGE Physical layer on the radio path; General description (Release 14) (3GPP). Technical report, 2017.

[71] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The Locator/ID Separation Protocol (LISP) (RFC 6830). 2013.

[72] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia. A Generic Quantitative Relationship Between Quality of Experience and Quality of Service. *IEEE Network*, 24(2), 2010.

[73] A. Ford, Costin Raiciu, Mark Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses (RFC 6824). 2013.

[74] V. Fuller and T. Li. Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan (RFC 4632). 2006.

[75] General Motors Company. GM OnStar: Get to know OnStar. Accessed at 2017-02-11. URL https://www.onstar.com/us/en/services/services.html.

[76] Fabio Giust, Antonio De Oliva, and Carlos J Bernardos. Flat Access and Mobility Architecture : an IPv6 Distributed Client Mobility Management Solution. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM Workshop)*, 2011.

[77] Fabio Giust, Carlos J. Bernardos, and Antonio De La Oliva. Analytic evaluation and experimental validation of a network-based IPv6 distributed mobility management solution. *IEEE Transactions on Mobile Computing*, 13(11), 2014.

[78] Fabio Giust, Carlos J Bernardos, and Antonio De Oliva. HDMM: Deploying Client and Network-Based Distributed Mobility - A Hybrid Approach. *Springer Telecommunication Systems*, 59(2), 2015.

[79] Fabio Giust, Luca Cominardi, and Carlos J Bernardos. Distributed Mobility Management for Future 5G Networks : Overview and Analysis of Existing Approaches. *IEEE Communications Magazine*, 53(1), 2015.

[80] Fernando Gont. Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery (RFC 6980). 2013.

[81] J. Gozalvez, M. Sepulcre, and R. Bauza. IEEE 802.11p vehicle to infrastructure communications in urban environments. *IEEE Communications Magazine*, 50 (5), 2012.

[82] Kálmán Graffi. *Monitoring and Management of Peer-to-Peer Systems*. PhD thesis, Technische Universität Darmstadt, 2011.

[83] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6 (RFC 5213). 2008.

[84] Guolin Sun, Guisong Liu, Hangming Zhang, Wei Tan, and Wei Tan. Architecture on Mobility Management in OpenFlow- based Radio Access Networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2013.

[85] Vivik Gupta, Subir Das, Anthony Chan, David Cypher, and Et al. IEEE Standard for Local and metropolitan area networks–Media Independent Handover Services (IEEE 802.21). Technical report, 2008.

[86] Guillaume Habault, Laurent Toutain, Nicolas Montavont, and Philippe Bertin. Service-Based Network Selection Proposal for Complex Heterogeneous Environments. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2014.

[87] Mark Handley and Rolf Winter. *Implementation and Assessment of Modern Host-based Multipath Solutions*. PhD thesis, Louvain School of Engineering, Université Catholique de Louvain, Belgium, 2011.

[88] Ronny Hans, David Steffen, Ulrich Lampe, Dominik Stingl, and Ralf Steinmetz. Short Run : Heuristic Approaches for Cloud Resource Selection. In *Proceedings of the IEEE International Conference on Cloud Computing (CLOUD)*, 2016.

[89] Peter Hart, Nils Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 1968.

[90] Brandon Heller, Rob Sherwood, and Nick McKeown. The Controller Placement Problem. In *Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Software Defined Networks (HotSDN)*, New York, New York, USA, 2012. ACM Press.

[91] Joachim Hillebrand, Christian Prehofer, Roland Bless, and Martina Zitterbart. Quality-of-Service Signaling for Next Generation , IP-based Mobile Networks. *Springer Wireless Personal Communications*, 43(3), 2007.

[92] James F. Hines. Forecast: Connected Car Production, Worldwide. Technical report, Gartner Inc., 2016.

[93] Matthias Hollick. *Dependable Routing for Cellular and Ad-hoc Networks*. PhD thesis, Technische Universität Darmstadt, 2004.

[94] Melanie Holloway. *Service Level Management in Cloud Computing - Pareto-Efficient Concurrent Multiple-Issue Negotiations, Reliable Consumer-Side Availability Monitoring and Strategies for Robust Monitor Placement*. PhD thesis, 2016.

[95] IEEE. 802.11n-2009 - IEEE Standard for Information technology- Local and metropolitan area networks- Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. Technical report, IEEE, 2009.

[96] IEEE. 802.11p-2010 - IEEE Standard for Information Technology- Local and metropolitan area networks- Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Envir, 2010.

[97] IEEE. 802.11ac-2013 - IEEE Standard for Information technology- Telecommunications and information exchange between systems - Local and metropolitan area networks- Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY), 2013.

[98] IEEE. 802.11ah-2016 - IEEE Standard for Information technology- Telecommunications and information exchange between systems - Local and metropolitan area networks- Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY. Technical report, IEEE, 2016.

[99] Youngbin Im and Carlee Joe-wong. AMUSE : Empowering Users for Cost-Aware Offloading with Throughput-Delay Tradeoffs. *IEEE Transactions on Mobile Computing*, 15(5):1062–1076, 2016.

[100] Sofiane Imadali, Arnaud Kaiser, Fikret Sivrikaya, Nadim El Sayed, Michael Boc, Witold Klaudel, Alexandru Petrescu, and Veronique Veque. A Review of Network Mobility Protocols for Fully Electrical Vehicles Services. *IEEE Intelligent Transportation Systems Magazine*, 6(3):80–95, 2014.

[101] ISO. EN 21217:2014, Intelligent transport systems - Communications Access for Land Mobiles (CALM) - Architecture. Technical report, 2014.

[102] Florian Jomrich. Intelligent Provisioning of High Definition Street Maps for Highly Automated Driving Vehicles. In *Proceedings of the International Conference on Networked Systems (NetSys)*, 2017.

[103] Florian Jomrich, Tobias Rückelt, Doreen Böhnstedt, and Ralf Steinmetz. An Efficient Heat-Map-Based Wireless Communication Simulation Model for Omnet++. In *Proceedings of the Automotive meets Electronics (AME)*, 2017.

[104] Florian Jomrich, Aakash Sharma, Tobias Rückelt, Daniel Burgstahler, and Doreen Böhnstedt. Dynamic Map Update Protocol for Highly Automated Driving Vehicles. In *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 2017.

[105] Florian Jormrich. *Seamless Handover in Wireless Vehicular Communication Networks*. Master thesis, Technische Universität Darmstadt, 2015.

[106] Fabian Kaup, Foivos Michelinakis, Nicola Bui, Joerg Widmer, Katarzyna Wac, and David Hausheer. Behind the NAT - A Measurement Based Evaluation of Cellular Service Quality. In *In Proceedings of the IEEE International Conference on Network and Service Management (CNSM)*, 2015.

[107] Lutz Kelch, Tobias Pogel, Lars Wolf, and Andreas Sasse. CQI Maps for Optimized Data Distribution. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*, 2013.

[108] S. Kent. IP Authentication Header (RFC 4302). 2005.

[109] S. Kent. IP Encapsulating Security Payload (ESP) (RFC 4303). 2005.

[110] Markus Kerper, Christian Wewetzer, Andreas Sasse, and Martin Mauve. Learning Traffic Light Phase Schedules from Velocity Profiles in the Cloud. In *Proceedings of the International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2012.

[111] Manzoor Ahmed Khan and Umar Toseef. User Utility Function as Quality of Experience (QoE). In *Proceedings of the International Conference on Networks (ICN)*, 2011.

[112] Manzoor Ahmed Khan, Umar Toseef, Stefan Marx, and Carmelita Goerg. Game-Theory Based User Centric Network Selection with Media Independent Handover Services and Flow Management. In *Proceedings of the IEEE Annual Conference on Communication Networks and Services Research (CNSR)*, 2010.

[113] Manzoor Ahmed Khan, Sebastian Peters, Xuan Thuy Dang, Tobias Dörsch, and Patrick Engelhard. On the Approaches for Efficient Mobility Management in SDNized Wireless Networks. In *Proceedings of the IEEE International Conference on Communication and Networking (Comnet)*, 2017.

[114] D. P. Kim and S. J. Koh. Analysis of Handover Latency for Mobile IPv6 and mSCTP. In *Proceedings of the IEEE International Conference on Communications (ICC Workshops)*, 2008.

[115] Hongseok Kim, Gustavo de Veciana, Xiangying Yang, and Muthaiah Venkatachalam. Distributed Alpha-Optimal User Association and Cell Load Balancing in Wireless Networks. *IEEE/ACM Transactions on Networking*, 20(1): 177–190, 2012.

[116] Christian Koch and David Hausheer. Optimizing Mobile Prefetching by Leveraging Usage Patterns and Social Information. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2014.

[117] S.J. Koh, M.J. Chang, and M. Lee. mSCTP for Soft Handover in Transport Layer. *IEEE Communications Letters*, 8(3):189–191, 2004.

[118] Evangelia Kokolaki, Merkouris Karaliopoulos, and Ioannis Stavrakakis. Opportunistically Assisted Parking Service Discovery: Now It Helps, Now It Does Not. *Elsevier Pervasive and Mobile Computing*, 8(2):210–227, 2012.

[119] Evangelia Kokolaki, Merkouris Karaliopoulos, and Ioannis Stavrakakis. Leveraging Information in Parking Assistance Systems. *IEEE Transactions on Vehicular Technology*, 62(9):4309–4317, 2013.

[120] Paul Kompfner. Cooperative Vehicle Infrastructure Systems (CVIS), ACM Sigmobile Keynote Presentation. Accessed at 2017-05-15, 2007. URL https://www.sigmobile.org/workshops/vanet2007/slides/Kompfner.pdf.

[121] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication (RFC 2104). 1997.

[122] Ying-Feng Kuo, Chi-Ming Wu, and Wei-Jaw Deng. The Relationships Among Service Quality, Perceived Value, Customer Satisfaction, and Post-Purchase Intention in Mobile Value-Added Services. *Computers in Human Behavior*, 25 (4), 2009.

[123] Stanislav Lange, Steffen Gebert, Thomas Zinner, Phuoc Tran-Gia, David Hock, Michael Jarschel, and Marco Hoffmann. Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks. *IEEE Transactions on Network and Service Management*, 12(1), 2015.

[124] Bob Lantz, Brandon Heller, and Nick McKeown. A Network in a Laptop. *Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, 2010.

[125] N. Laoutaris and I. Stavrakakis. Adaptive PPlayout Strategies for Packet Video Receivers with Finite Buffer Capacity. In *Proceedings of the IEEE International Conference on Communications. Conference Record (ICC)*, 2001.

[126] Franck Le, Geoffrey G. Xie, and Hui Zhang. On Route Aggregation. In *Proceedings of the ACM COnference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2011.

[127] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the ACM Conference on Multimedia Systems (MMSys)*. ACM Press, 2012.

[128] Jong-Hyouk Lee, Thierry Ernst, and Naveen Chilamkurti. Performance Analysis of PMIPv6-Based NEtwork MObility for Intelligent Transportation Systems. *IEEE Transactions on Vehicular Technology*, 61(1):74–85, 2012.

[129] Jong-hyouk Lee, Jean-marie Bonnin, Senior Member, and Ilsun You. Comparative Handover Performance Analysis of IPv6 Mobility Management Protocols. *IEEE Transactions on Industrial Electronics*, 60(3), 2013.

[130] Jong Hyouk Lee, Jean Marie Bonnin, Pierrick Seite, and H. Anthony Chan. Distributed IP Mobility Management from the Perspective of the IETF: Motivations, Requirements, Approaches, Comparison, and Challenges. *IEEE Wireless Communications*, 20(5), 2013.

[131] Joohyun Lee, Yung Yi, Song Chong, and Youngmi Jin. Economics of WiFi Offloading: Trading Delay for Cellular Capacity. *IEEE Transactions on Wireless Communications*, 13(3):1540–1554, 2014.

[132] Joohyun Lee, Kyunghan Lee, Choongwoo Han, Taehoon Kim, and Song Chong. Resource-Efficient Mobile Multimedia Streaming With Adaptive Network Selection. *IEEE Transactions on Multimedia*, 18(12), 2016.

[133] Kyunghan Lee, Associate Member, Joohyun Lee, Student Member, and Yung Yi. Mobile Data Offloading : How Much Can WiFi Deliver ? *Proceedings of the ACM COnference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2010.

[134] Dan Li. *In-situ Communication Schedule Adaptation for Vehicles*. Master thesis, Technische Universität Darmstadt, 2016.

[135] Yao Liu, Sujit Dey, Don Gillies, Faith Ulupinar, and Michael Luby. User Experience Modeling for DASH Video. In *Proceedings of the IEEE International Packet Video Workshop (IPV)*. IEEE, 2013.

[136] Inc LoRa Alliance. LoRa Specification. Technical report, 2015.

[137] Zheng Lu and Gustavo De Veciana. Optimizing Stored Video Delivery for Mobile Networks : The Value of Knowing the Future. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2013.

[138] Sassi Maaloul, Meriem Afif, and Sami Tabbane. An efficient handover decision making for heterogeneous wireless connectivity management. In *IEEE International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013)*, 2013.

[139] Machina Research. M2M Growth Necessitates a New Approach to Network Planning and Optimisation. Technical Report May, 2015.

[140] Andreas Mai. Connected Vehicles: From Building Cars to Selling Personal Travel Time Well-Spent. Technical report, Cisco, 2011.

[141] Spyros Makridakis. Accuracy Measures: Theoretical and Practical Concerns. *International Journal of Forecasting*, 9(4), 1993.

[142] Ilaria Malanchini, Matteo Cesana, and Nicola Gatti. Network Selection and Resource Allocation Games for Wireless Access Networks. *IEEE Transactions on Mobile Computing*, 12(12), 2013.

[143] G.B Mathews. On the Partition of Numbers. In *Proceedings of the London Mathematical Society*, 1897.

[144] Fidan Mehmeti and Thrasyvoulos Spyropoulos. Is it Worth to Be Patient? Analysis and Optimization of Delayed Mobile Data Offloading. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2014.

[145] Fidan Mehmeti and Thrasyvoulos Spyropoulos. Performance Modeling, Analysis, and Optimization of Delayed Mobile Data Offloading for Mobile Users. *IEEE/ACM Transactions on Networking*, 25(1), 2016.

[146] Tobias Meuser, Daniel Burgstahler, Tobias Rückelt, Doreen Böhnstedt, and Ralf Steinmetz. Hybrid-ProbSense.KOM: Probabilistic Sensing with Hybrid Communication for Gathering Vehicular Sensed Data. In *Proceedings of the Automotive meets Electronics (AmE)*, 2017.

[147] John E. Mitchell. Branch-and-Cut Algorithms for Combinatorial Optimization Problems. In *Handbook of Applied Optimization*. Oxford University Press, 2000.

[148] D Mohr, N Muller, A Krieg, P Gao, H W Kaas, A Krieger, and R Hensley. The road to 2020 and beyond: What's driving the global automotive industry? Technical report, 2013.

[149] N. Moore. Optimistic Duplicate Address Detection (DAD) for IPv6. 2006.

[150] Danny Moses, Muthaiah Venkatachalam, and Intel Corporation. Distributed Mobility Management for Efficient Video Delivery over All-IP Mobile Networks: Competing Approaches. *IEEE Nework*, 27(2), 2013.

[151] R Moskowith, T. Heer, P. Jokela, and T. Henderson. Host Identity Protocol Version 2 (HIPv2) (RFC 7401). 2015.

[152] J. Moy. Open Shortest Path First Version 2 (OSPF) (RFC 2328). 1998.

[153] L. Srikar Muppirisetty, John Tadrous, Atilla Eryilmaz, and Henk Wymeersch. On Proactive Caching with Demand and Channel Uncertainties. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015.

[154] Ghulam Murtaza, Andreas Reinhardt, Mahbub Hassan, and Salil S Kanhere. Creating Personal Bandwidth Maps using Opportunistic Throughput Measurements. In *Proceedins of the IEEE International Conference on Communications (ICC)*, 2014.

[155] Robert Nagel and Stefan Morscher. Connectivity Prediction in Mobile Vehicular Environments Backed By Digital Maps. In *Advanced Trends in Wireless Communications*, chapter 13, pages 243–264. InTech, 2011. ISBN 978-953- 307-183-1.

[156] Sinh Chung Nguyen, Xiaofei Zhang, Thi Mai Trang Nguyen, and Guy Pujolle. Evaluation of Throughput Optimization and Load Sharing of Multipath TCP in Heterogeneous Networks. In *Proceedings of the International Conference on Wireless and Optical Communications Networks (WOCN)*, 2011. ISBN 978-1-4577-0262-4. doi: 10.1109/WOCN.2011.5872966.

[157] Catalin Nicutar, Christoph Paasch, Marcelo Bagnulo, Costin Raiciu, and U Politehnica Bucharest. Evolving the Internet with Connection Acrobatics. 2013.

[158] Ashkan Nikravesh, Yihua Guo, Feng Qian, Z Morley Mao, and Subhabrata Sen. An In-Depth Understanding of Multipath TCP on Mobile Devices. *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2016.

[159] E. Nordmark and M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6 (RFC 5533). Technical report, 2009.

[160] Richard L. Oliver. A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions. *Journal of Marketing Research*, 17(4), 1980.

[161] Olga Ormond and John Murphy. Utility-based Intelligent Network Selection in Beyond 3G Systems. In *IEEE International Conference on Communications (ICC)*, 2006.

[162] Ae-Soon Park and S. Venkatesan. Handover Prediction Strategy for 3G-WLAN Overlay Networks. In *IEEE Network Operations and Management Symposium (NOMS)*. Ieee, 2008.

[163] Xavier Pérez-Costa, Marc Torrent-Moreno, and Hannes Hartenstein. A performance comparison of Mobile IPv6, Hierarchical Mobile IPv6, fast handovers for Mobile IPv6 and their combination. *ACM SIGMOBILE Mobile Computing and Communications*, 7(4), 2003.

[164] C. Perkins, D. Johnson, and J. Arkko. Mobility Support in IPv6 (RFC 6275). 2011.

[165] Henrik Petander and Eranga Perera. Measuring and Improving the Performance of Network Mobility Management in IPv6 Networks. *IEEE Journal on Selected Areas in Communications*, 24(9), 2006.

[166] Tobias Poegel and Lars Wolf. Optimization of Vehicular Applications and Communication Properties with Connectivity Maps. In *Proceedings of the IEEE Local Computer Networks Conference Workshops (LCN)*, 2015.

[167] Konstantin Pussep. *Peer-Assisted Video-on-Demand: Cost Reduction and Performance Enhancement for Users, Overlay Providers, and Network Operators*. PhD thesis, Technische Universität Darmstadt, 2011.

[168] Mr Moo-Ryong Ra, Jeongyeup Paek, Abhishek B. Sharma, Ramesh Govindan, Martin H. Krieger, and Michael J. Neely. Energy-Delay Tradeoffs in

Smartphone Applications. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and ServicesMobiSys (MobiSys)*, 2010.

[169] Md. Sazzadur Rahman and Mohammed Atiquzzaman. SEMO6 - A Multihoming-Based Seamless Mobility Management Framework. In *In Proceedings of the IEEE Military Communications Conference (MILCOM(*, 2008.

[170] Md Sazzadur Rahman, Mohammed Atiquzzaman, Wesley Eddy, and William Ivancic. Performance Comparison between MIPv6 and SEMO6. *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2010.

[171] Costin Raiciu, Dragos Niculescu, Marcelo Bagnulo, and Mark James Handley. Opportunistic Mobility with Multipath TCP. In *Proceedings of the ACM International Workshop on MobiArch*, 2011.

[172] K R Rao, Zoran S Bojkovic, and Bojan M Bakmaz. Network Selection in Heterogeneous Environment : A Step toward Always Best Connected and Served. In *Proceedings of the IEEE International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, 2013.

[173] Ulrich Reiter, Kjell Brunnström, Katrien De Moor, Mohamed-Chaker Larabi, Manuela Pereira, Antonio Pinheiro, Junyong You, and Andrej Zgank. Factors Influencing Quality of Experience. In *Quality of Experience*, pages 55–72. Springer International Publishing, 2014. doi: 10.1007/978-3-319-02681-7_4.

[174] Nicolas Repp. *Überwachung und Steuerung dienstbasierter Architekturen - Verteilungsstrategien und deren Umsetzung*. PhD thesis, Technische Universität Darmstadt, 2009.

[175] E. Rescorla. The Transport Layer Security (TLS) Protocol, Version 1.2 (RFC 5246). 2008.

[176] Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe, and Ralf Steinmetz. Seamless Transitions between Filter Schemes for Location-Based Mobile Applications. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*. IEEE, 2016.

[177] Nils Richerzhagen, Tao Li, Dominik Stingl, Björn Richerzhagen, Ralf Steinmetz, and Silvia Santini. A step towards a protocol-independent measurement framework for dynamic networks. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*. IEEE, 2015.

[178] Nils Richerzhagen, Dominik Stingl, Björn Richerzhagen, Andreas Mauthe, and Ralf Steinmetz. Adaptive Monitoring for Mobile Networks in Challenging Environments. In *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015.

[179] Lena Rittger, Gerald Schmidt, Christian Maag, and Andrea Kiesel. Driving Behaviour at Traffic Light Intersections. *Springer/ACM Journal on Cognition, Technology & Work*, 17(4), 2015.

[180] Tobias Rückelt and Carsten Büttner. Sicheres Service Management für ITS-Dienste. In *Proceedings of the Automotive meets Electronics (AmE)*, 2015.

[181] Julius Rückert. *Large-scale Live Video Streaming over the Internet - Efficient and Flexible Content Delivery Using Network and Application-Layer Mechanisms*. PhD thesis, Technische Universität Darmstadt, 2016.

[182] T. Rueckelt, H. Altug, D. Burgstahler, D. Böhnstedt, and R. Steinmetz. MoVe-Net: Mobility Management for Vehicular Networking. In *Proceedings of the ACM International Symposium on Mobility Management and Wireless Access (MobiWAC), co-located with MSWiM.*, 2016.

[183] T. Rueckelt, D. Burgstahler, F. Jomrich, D. Böhnstedt, and R. Steinmetz. Impact of Time in Network Selection for Mobile Nodes. In *Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2016.

[184] Tobias Rueckelt, Daniel Burgstahler, Frank Englert, Christian Gottron, and Sebastian Zöller. A Concept for Vehicle Internet Connectivity for Non-Safety Applications. In *Proceedings of the IEEE Local Computer Networks Conference (LCN)*, 2014.

[185] Tobias Rueckelt, Florian Jomrich, Daniel Burgstahler, Doreen Böhnstedt, and Ralf Steinmetz. Publish-Subscribe-Based Control Mechanism for Scheduling Integration in Mobile IPv6. In *Proceedings of the IEEE Local Computer Networks Conference (LCN)*, 2015.

[186] Almudena Sánchez, Rocío Acedo-Hernández, Matías Toril, Salvador Luna-Ramírez, Carlos Úbeda, Mated Toril, Luna-Ramed, Salvador Rez, and Carlos Beda. A Trace Data-Based Approach for an Accurate Estimation of Precise Utilization Maps in LTE. *Mobile Information Systems*, 2017.

[187] Sandvine. Global Internet Phenomena Asia-Pacific & Europe. Technical report, 2015.

[188] Jose Santa, Fernando Pereniguez-Garcia, Fernando Bernal, Pedro J. Fernandez, Rafael Marin-Lopez, and Antonio F. Skarmeta. A Framework for Supporting Network Continuity in Vehicular IPv6 Communications. *IEEE Intelligent Transportation Systems Magazine*, 6(1):17–34, 2014.

[189] Steffen Schnitzer, Sebastian Schmidt, Christoph Rensing, and Bettina Harriehausen-Mühlbauer. Combining Active and Ensemble Learning for Efficient Classification of Web Documents. *Polibits: Research Journal on Computer Science and Computer Engineering with Applications*, 49(7), 2014.

[190] Mohamed Abdelkrim Senouci, Mustapha Reda Senouci, Said Hoceini, and Abdelhamid Mellouk. An Evidential Approach for Network Interface Selection in Heterogeneous Wireless Networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2016.

[191] Samar Shailendra, R. Bhattacharjee, and Sanjay K. Bose. MPSCTP: A Multipath Variant of SCTP and its Performance Comparison with other Multipath Protocols. In *Proceedings of the IEEE International Conference on Communication Systems (ICCS)*. IEEE, nov 2012.

[192] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-Free Replicated Data Types. In *Proceedings of the Symposium on Self-Stabilizing Systems*. Springer Berlin Heidelberg, 2011.

[193] SimTD Consortium. Safe and Intelligent Mobility Test Field Germany (SimTD). Technical report, 2009.

[194] N Singh and Brahmjit Singh. Proxy Mobile IPv6-Based Handovers for VoIP Services in Wireless Heterogeneous Networks. 4(5), 2012. doi: 10.7763/IJET.2012.V4.425.

[195] Alexander Skabardonis and Nikolaos Geroliminis. Transportation and Traffic Theory: Flow, Dynamics and Human Interaction. In *Proceedings of the International Symposium on Transportation and Traffic Theory*, 2005.

[196] H. Soliman, K. ElMalki, and L. Bellier. Hierarchical Mobile IPv6 (HMIPv6) Mobility Management (RFC 5380). 2008.

[197] Kyuho Son, Song Chong, and Gustavo Veciana. Dynamic association for load balancing and interference avoidance in multi-cell networks. *IEEE Transactions on Wireless Communications*, 8(7), 2009.

[198] Nicolas Staelens, Stefaan Moens, Wendy Van den Broeck, Ilse Marien, Brecht Vermeulen, Peter Lambert, Rik Van de Walle, and Piet Demeester. Assessing Quality of Experience of IPTV and Video on Demand Services in Real-Life Environments. *IEEE Transactions on Broadcasting*, 56(4), 2010.

[199] Ralf. Steinmetz. *Multimedia-Technologie : Grundlagen, Komponenten und Systeme*. Springer-Verlag Berlin Heidelberg, 2000. ISBN 3540673326.

[200] Ralf Steinmetz and Klara Nahrstedt. *Multimedia Applications*. Springer-Verlag Berlin Heidelberg, 2004. ISBN 978-3-07410-3.

[201] Ralf Steinmetz and Klara Nahrstedt. *Multimedia : Computing, Communications and Applications*. Pearson, 2012. ISBN 8177584413.

[202] Dominik Stingl. *Decentralized Monitoring in Mobile Ad Hoc Networks - Provisioning of Accurate and Location-Aware Monitoring Information*. PhD thesis, Technische Universität Darmstadt, 2014.

[203] Tarik Taleb, Senior Member, Adlen Ksentini, and Senior Member. VECOS : A Vehicular Connection Steering Protocol. 64(3):1171–1187, 2015.

[204] Robert Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2), 1971.

[205] Tesla Inc. EVE For Tesla: Premier Dashboard Experience. Accessed at 2017-02-11. URL https://teslaapps.net/en/eveconnect/.

[206] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration (RFC 4862). 2007.

[207] Daxin Tian, Jianshan Zhou, Yunpeng Wang, Yingrong Lu, Haiying Xia, and Zhenguo Yi. A Dynamic and Self-Adaptive Network Selection Method for Multimode Communications in Heterogeneous Vehicular Telematics. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 2015.

[208] Ramona Trestian, Olga Ormond, and Gabriel-Miro Muntean. Game Theory-Based Network Selection: Solutions and Challenges. *IEEE Communications Surveys & Tutorials*, 14(4), 2012.

[209] G. Tsirtsis, H. Soliman, Nicolas Montavont, G. Giaretta, and K. Kuladinithi. Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support (RFC 6089), 2011.

[210] S. Turner and L. Chen. Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms (RFC 6151). 2011.

[211] Volkswagen AG. VW Car-Net: Apps und Dienste. Accessed at 2017-02-11. URL http://volkswagen-carnet.com/de/de/start/app-overview.html{#}tab/open/vw-cat-app-connect.

[212] Hao Wang, David I Laurenson, and Jane Hillston. Framework for Network Selection Strategies in 3G-WLAN Interworking Networks. *IEEE Transactions on Mobile Computing*, 12(5):868–884, 2012.

[213] Jidong Wang. HIP Based Mobility Management for UMTS / WLAN Integrated Networks. In *In Proceedings of the IEEE Australian Telecommunication Networks & Applications Conference*, 2006.

[214] Lusheng Wang and Geng-Sheng G.S. Kuo. Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks - A Tutorial. *IEEE Communications Surveys & Tutorials*, 15(1), 2013.

[215] Jakob Weimar. *Analysis of TCP Behavior during Transparent Handover of MoVeNet*. Lab report, Technische Universität Darmstadt, 2016.

[216] Matthias Wichtlhuber, Björn Richerzhagen, Julius Rückert, and David Hausheer. TRANSIT: Supporting transitions in Peer-to-Peer live video streaming. In *Proceedings of the IFIP Networking Conference*. IEEE, 2014.

[217] Qihui Wu, Senior Member, Zhiyong Du, Student Member, and Panlong Yang. Traffic-Aware Online Network Selection in Heterogeneous Wireless Networks. *IEEE Transactions on Vehicular Technology*, 65(1), 2016.

[218] Yang Xu, Chenguang Yu, Jingjiang Li, and Yong Liu. Video Telephony for End-Consumers. In *Proceedings of the ACM conference on Internet Measurement Conference (IMC)*, 2012.

[219] Zichuan Xu, Weifa Liang, Wenzheng Xu, Mike Jia, and Song Guo. Capacitated Cloudlet Placements in Wireless Metropolitan Area Networks. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*. IEEE, 2015.

[220] Guang Yao, Jun Bi, Yuliang Li, and Luyi Guo. On the Capacitated Controller Placement Problem in Software Defined Networks. *IEEE Communications Letters*, 18(8), 2014.

[221] J. Yu. Scalable Routing Design Principles (RFC 2791). 2000.

[222] Said Zaghloul, Admela Jukan, and Wesam Alanqar. Extending QoS from Radio Access to an All-IP Core in 3G Networks: An Operator's Perspective. *IEEE Communications Magazine*, 45(9), 2007.

[223] Kun Zhu, Dusit Niyato, and Ping Wang. Network Selection in Heterogeneous Wireless Networks : Evolution with Incomplete Information. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2010.

[224] Xuejun Zhuo, Wei Gao, Guohong Cao, and Yiqi Dai. Win-Coupon : An Incentive Framework for 3G Traffic Offloading. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2011.

[225] J.G. Ziegler and N.B. Nichols. Optimum Settings for Automatic Controllers. *Transactions of the American Society of Mechanical Engineers*, 64, 1942.

[226] Juan Carlos Zuniga, Carlos J. Bernardos, Antonio De La Oliva, Telemaco Melia, Rui Costa, and Alex Reznik. Distributed Mobility Management: A Standards Landscape. *IEEE Communications Magazine*, 51(3), 2013.

A APPENDIX

A.1 COMPLEXITY ANALYSIS OF THE TRANSMISSION PLANNING PROBLEM

In the following, we analyze the complexity of transmission planning problem and show that it is NP-hard due to the equivalence of a function component to multiple-demand bounded multiple knapsack problem. Problem definition:

Given a transmission plan p*, a set of networks* N *and a set of data flows* F*, is* p *a feasible transmission plan and is* p *of minimal cost?*

Firstly, the transmission rating model in Section 4.1 defines a polynomial-time cost function $c(p)$ in Equation 4.4 with polynomial-time constraints C1-C4. Accordingly, the feasibility and cost value of a transmission plan p can be determined in polynomial time, defining Lemma 1 (L1).

Secondly, we consider only a sub-problem. The cost of unscheduled tokens $c_f^u(p)$ according to equation 4.9 takes effect independently from time and depends on individual data flow parameters. Referring to the multiple-demand bounded multiple knapsack problem, tokens of different data flows with their individual cost $c_f^u(p)$ represent multiple, different demands. Furthermore, tokens are indivisible and their number is bounded. Thus, they and are equivalent to the bounded number of equal items. Limited knapsack sizes are represented by limited network resources, whereby multiple networks with individual capacity exist. Hence, this sub-problem of the transmission planning problem is equivalent to the specialized knapsack problem, defining Lemma 2 (L2). According to this equivalence, it is not possible to verify in polynomial time whether p is of minimal cost or not, given $P \neq NP$.

As transmission planning is not in NP (L1) and the knapsack problem is a sub-problem of transmission planning that is NP-complete (L2), the transmission planning problem is NP-hard.

A.2 RANDOM TRANSMISSION PLANNING ALGORITHM

Transmission planning algorithms have to satisfy the constraints of the rating model. Hence, constraint satisfaction is a basic requirement ensuring that the rating model, as defined in 4.1, is applicable and plans comply with the reality. For example, they must not plan to use more network resources than available.

The following Random Transmission Planning Algorithm guarantees to satisfy this requirement and tends to switch networks only if required. It sorts data flows randomly for token assignment, ensured by the shuffle function in line 3. For each time step in the time span between flow start time and deadline, it attempts to assign tokens to a randomly chosen network. If token assignment fails, the algorithm tries to assign the next randomly chosen network, excluding the ones already checked in this time step. In the case of success, it retains the network selection for the next time slot with a probability of P_{retain}. For evaluation, we selected $P_{retain} = 0.9$. This ensures, that network switches may happen without a force but do not happen in each time slot, leading to exceptionally high network migration cost.

Algorithm 4 RandomTransmissionPlanning

```
procedure TransmissionPlanning(flows, networks)
    plan ← empty plan
    flows ← shuffle(flows)                                  ▷ Shuffle data flows randomly
    networks ← shuffle(networks)                            ▷ Shuffle networks randomly
    for flow in flows do
        n ← 0                                               ▷ Index of network to assign to
        for time in flow.starttime to flow.deadline do
            plan_temp ← plan
            count ← networks.size()
            while count > 0 do                  ▷ Stop if assignment failed for each network
                network ← networks.get(n)
                plan ← assignTokens(flow, time, network, plan)
                if plan ! = plan_temp then
                    break               ▷ End the time slot if assignment was successful
                ▷ If no token assignment was possible, try next random network
                n ← (n+1)mod networks.size()                        ▷ Increase index
                if n= 0 or random < $P_{retain}$ then
                    networks ← shuffle(networks)    ▷ Shuffle networks randomly
                count ← count−1
```

A.3 DATA FLOW TYPES

For simulation of transmission planning algorithms, we categorize 4 data traffic categories: Interactive, Conversational, Bufferable and Background. Their share of the overall data traffic follows the measured and predicted distribution of Cisco [140] and Sandvine [187] for mobile Internet traffic with minor adaption for the connected vehicle scenario. Vehicles will integrate more and more autonomous driving functions and driving efficiency features in the future [29, 118, 119]. Most of these features profit from external information [35, 33]. For example, autonomous driving profits from a highly detailed map, which covers details of the environment [102, 104]. Systems should continuously update this information and, as well, push locally collected sensor data to servers to keep the remote map up-to-date [36, 30, 146]. Hence, we increase the amount of background data from the Internet traffic reports traffic by 10% for the connected vehicle scenario in comparison to the reports above and prognoses on smartphones. In a real system, application data flows may be classified, using methods inspired from [4, 189] The categories zero to very high in Table 15 represent randomized integer values from truncated Gaussian distributions, parameterized as shown in 16. In the following, we describe the characteristics of the default data flow template and the four data traffic categories briefly.

A.3.1 *Default Data Flow Template*

We define a default data flow as a template which defines values for a minimum set of configuration and requirements. The other data flows are defined differentially to it, overwriting dedicated requirement definitions. The default data flow has no data and medium importance to transmit data ω^u. For scenario generation in our evaluation, we set the user preference ω^{user} to the randomized value medium, whereas in real systems, it should be initialized with a fixed value that can be influenced by the user. The minimum throughput requirement is deactivated by setting zero tokens to be transmitted within a maximum time window of the entire scenario time span T and zero importance. The maximum throughput constraint is deactivated as well, allowing to transmit all tokens of the data flow in a single time slot. Latency and Jitter may be very high without a violation, weighted by zero. Start time and deadline are set to zero (now), respectively beyond the planning time horizon, both weighted with zero by default.

A.3.2 *Interactive Data Flow*

We set amount of data of interactive data flows to 5% of the overall scenario traffic share [187] and define it as a series of short information requests with a small data amount for which an instant response is expected, according to [122, 200]. Hence, we model it with very high priority to transmit data ω_u, low minimum throughput requirements with flow continuity requirements and a start time and deadline with high importance, summarized in Table 15 .

A.3.3 *Conversational Data Flow*

Conversational data flows refer to speech and video telephony or other data transfer with human interaction, covering 15% of the overall scenario traffic share [187]. We define its requirements according to Xu et al. [218] and Steinmetz [200, 201] with high importance of allocating all data, very high importance to satisfy a continuous medium data rate, strong requirements for a low latency and jitter and very high importance to transmit data between the desired start time and deadline.

A.3.4 *Bufferable Data Flow*

The bufferable data flow corresponds, e.g. to video-on-demand, music-on-demand or other semi-soft continuous transmissions, defining the biggest share of 55% of data traffic, which is still expected to increase [187]. We model its requirements according to Staelens et al. [198] and Steinmetz et al. [200] with a medium importance to transmit all data, but a very strong minimum throughput requirement and allowed transmission in bursts, reflected by a very high minimum throughput window size in the model. This corresponds to adaptive rate transmission, in which the quality of the video is matched to the experienced network quality, like DASH (Dynamic Adaptive Streaming over HTTP) [127, 135, 167, 1]. Hence, a certain minimum throughput should be satisfied, while transmission of additional data, corresponding to only medium importance to transmit all data, increases the quality of the video. A start time should approximately be satisfied [10] with high importance while the deadline might be later, e.g. due to pauses [125].

A.3.5 *Background Data Flow*

For background data, we define two types, differing in the requirement if they have a timely deadline to be finished. The first has a deadline far beyond the planning time horizon and uses the default data flow definition, except low importance to transmit data. It corresponds to non-urgent data transmissions, e.g. software updates, cloud synchronization or map material update from non-local regions [102]. In our scenario reflects 50% of the background data traffic. The second type differs from the default data flow definition due to a set start time and a deadline, keeping the medium importance to transmit data and representing, e.g. data provision for soon required map data [33] or sensor data transfers [32].

Type	**Default**	**Interactive**	**Conversational**	**Bufferable**	**Background**
Data Amount*	0.00	0.05	0.15	0.55	0.25
ω^{u}	medium	very high	high	medium	medium
ω^{user}	medium[2]				
min. Throughput	zero	low	medium	high	
min. Window[3]	T	very low	very low	very high	
ω^{tp}	zero	medium	very high	very high	
max. Throughput	all tokens		medium		
max. Window[3]	one		very low		
req. Latency	very high	low	low		
ω^{lcy}	zero	medium	very high		
req. Jitter	very high		low		
ω^{jit}	zero		very high		
Start time	zero	st	st	st	st
ω^{st}	zero	very high	very high	high	low
Deadline	T	dl	dl	dl	(dl)
ω^{dl}	zero	very high	very high	medium	medium

[1] multiplied by the overall number of tokens and normalized by the number of data flows of the same type in the scenario

[2] default user preference should be a fixed value in real systems, which might be modified by the user [3] corresponds to the degree to which the transmission may be in bursts, detailed in Section 4.1.2.4

Table 15: Data flow parameter definition

Type	Mean	Minimum	Maximum	Standard deviation
zero	0	0	0	0
very low	2	1	3	1
low	3	1	5	1
medium	6	3	9	2
high	10	6	14	2
very high	15	10	20	2

Table 16: Randomized integer parameter settings from truncated Gaussian distributions

Table 17: NRS gains of JTP with varied parameter $c_{\mathtt{lim}}$ compared to default JTP

$c_{\mathtt{lim}}$	5	10	15	20
NRS gain over JTP	0.00045	0.00045	-0.012	-0.012

A.4 PARAMETER OPTIMIZATION OF JOINT TRANSMISSION PLANNING

In Section 4.3.3, we identified a parameter imbalance for Joint Transmission Planning. A Relative Detail Score (RDS) analysis revealed that the approach underrates the cost from non-allocated tokens while it overrates monetary cost. The two correlate directly because allocating additional tokens causes additional monetary cost for transmission. Thus, a parameter change could solve the identified imbalance. The balance between attracting forces, which foster token allocation, and repelling forces, e.g. monetary cost, in JTP is influenced from the parameter $c_{\mathtt{lim}}$. It is the limit of which the heuristically approximated difference between attracting and repelling forces is interpreted as sufficient to justify a token allocation. To investigate if the revealed imbalance is curable, we increase the parameter $c_{\mathtt{lim}}$ to allocate more tokens. We select $c_{\mathtt{lim}} = \{0, 5, 10, 15, 20\}$. Figure 55 shows the NRS and RDS values of JTP with the named parameter options.

The mean gains of the parametrized options over default JTP are additionally presented in Table 17 because they are barely noticeable. We observe a slight but statistically insignificant improvement of the results of 0.00045 for $c_{\mathtt{lim}}$ set to 5 and 10. However, for larger values of $c_{\mathtt{lim}}$, the result quality diminishes. More interestingly, we can observe how the RDS of the transmission planner changes. As desired, the score for unscheduled tokens falls for higher values of $c_{\mathtt{lim}}$ and the monetary cost score rises. The other parameters stay nearly unaffected. This parameter optimization shows that RDS analyses indicate weaknesses of strategies correctly. However, in this case, a parameter optimization brings no significant positive effect. The defect must be in the strategy itself. As the positive effects of this parameter optimization are statistically insignificant, we keep the initial parameter $c_{\mathtt{lim}} = 0$ for evaluation.

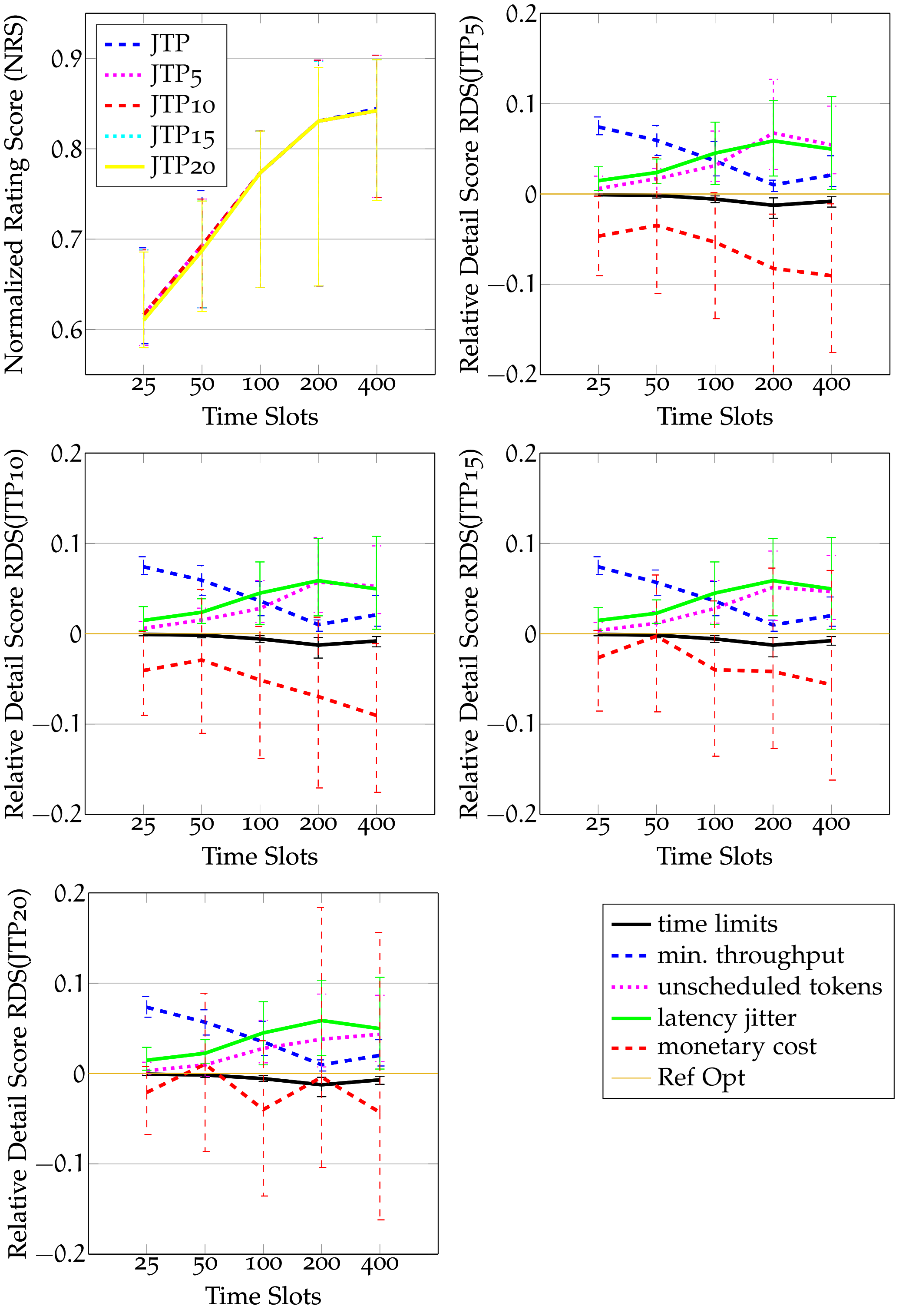


Figure 55: NRS and RDS for Joint Transmission Planning with varied parameter c_{lim} over number of time slots

A.5 FURTHER TRANSMISSION PLAN EVALUATION RESULTS

A.5.1 *Number of Networks*

The execution time of the heuristic transmission planners rises about proportionally with the number of networks. With maximum values below 0.1 seconds, they show sufficient performance for real system use. There is no significant difference between them. In contrast, the execution time of the optimal approach rises much faster than the heuristic approaches. The results are shown in Figure 56.

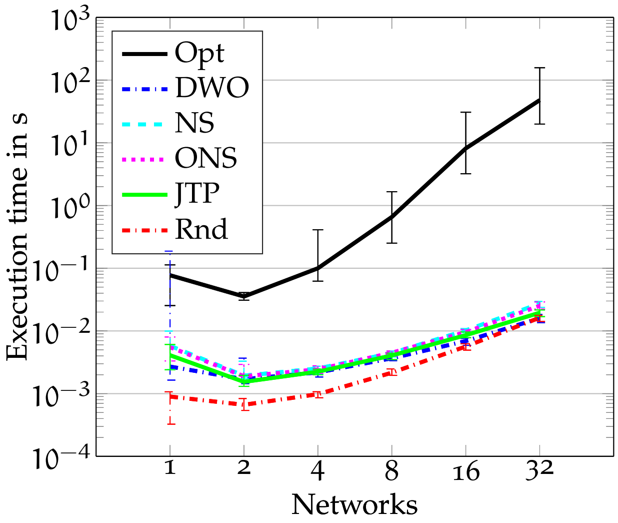


Figure 56: Execution duration over network number of networks

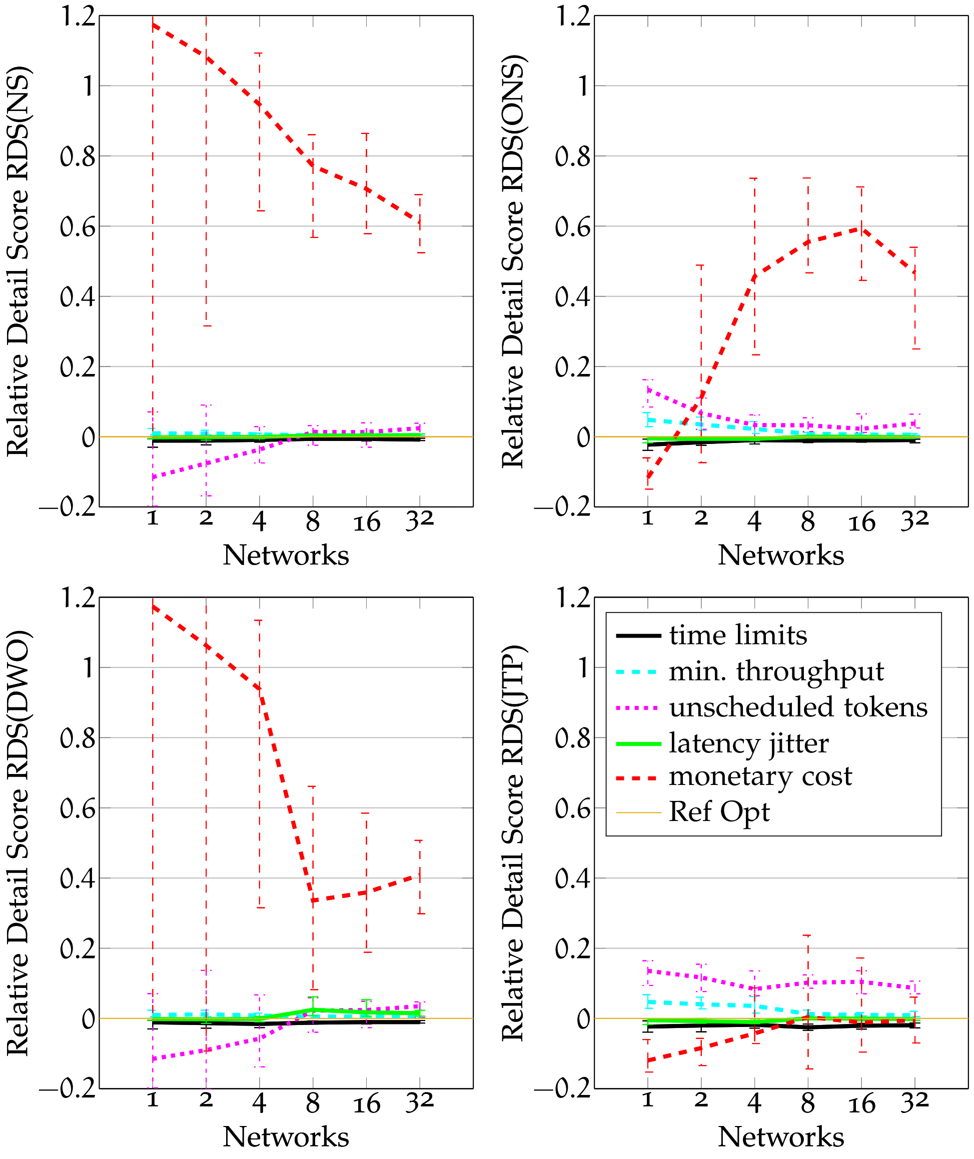


Figure 57: Relative Detail Scores of NS, ONS, DWO and JTP over number of networks

A.5.2 *Number of Data Flows*

The Relative Detail Scores in Figure 58 show sinking cost for monetary cost and widely stable values for the others. They cover the common characteristics, as discussed in the Strategic Transmission Planning evaluation in Section 4.3.

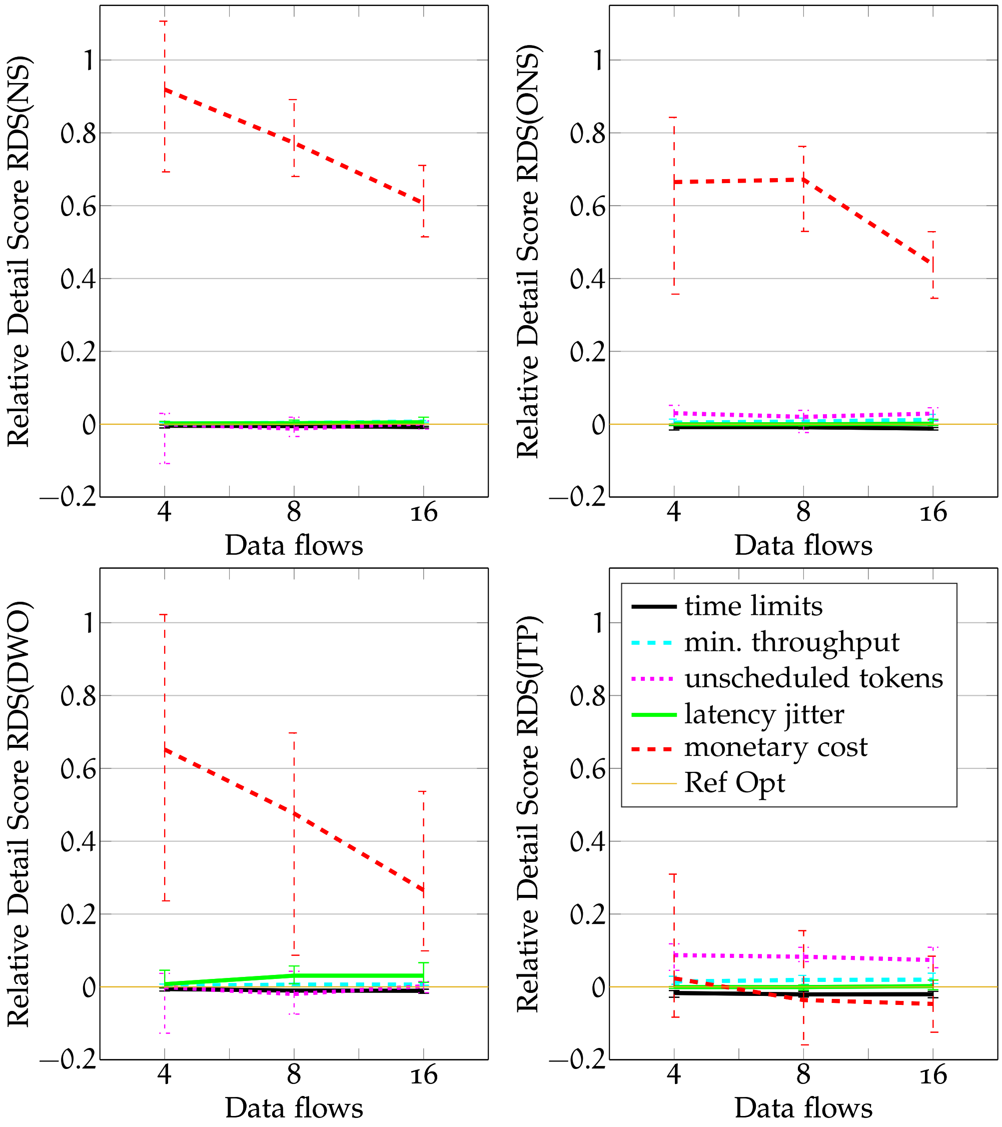


Figure 58: Relative Detail Scores of NS, ONS and JTP over the number of data flows

A.5.3 *Monetary Cost Weight*

The Relative Detail score over monetary cost weight confirms the finding from the corresponding Section 4.3.7. The monetary cost RDS value for Network Selection starts to explode for a medium and high weight. In contrast, it is still low for Opportunistic Network Selection and Joint Transmission Planning. The reason for this is, that both implement a heuristic to drop data or delay it beyond the planning horizon. The effects are clearly visible in Network Selection's unscheduled token RDS value, which gets below zero for a high monetary cost weight.

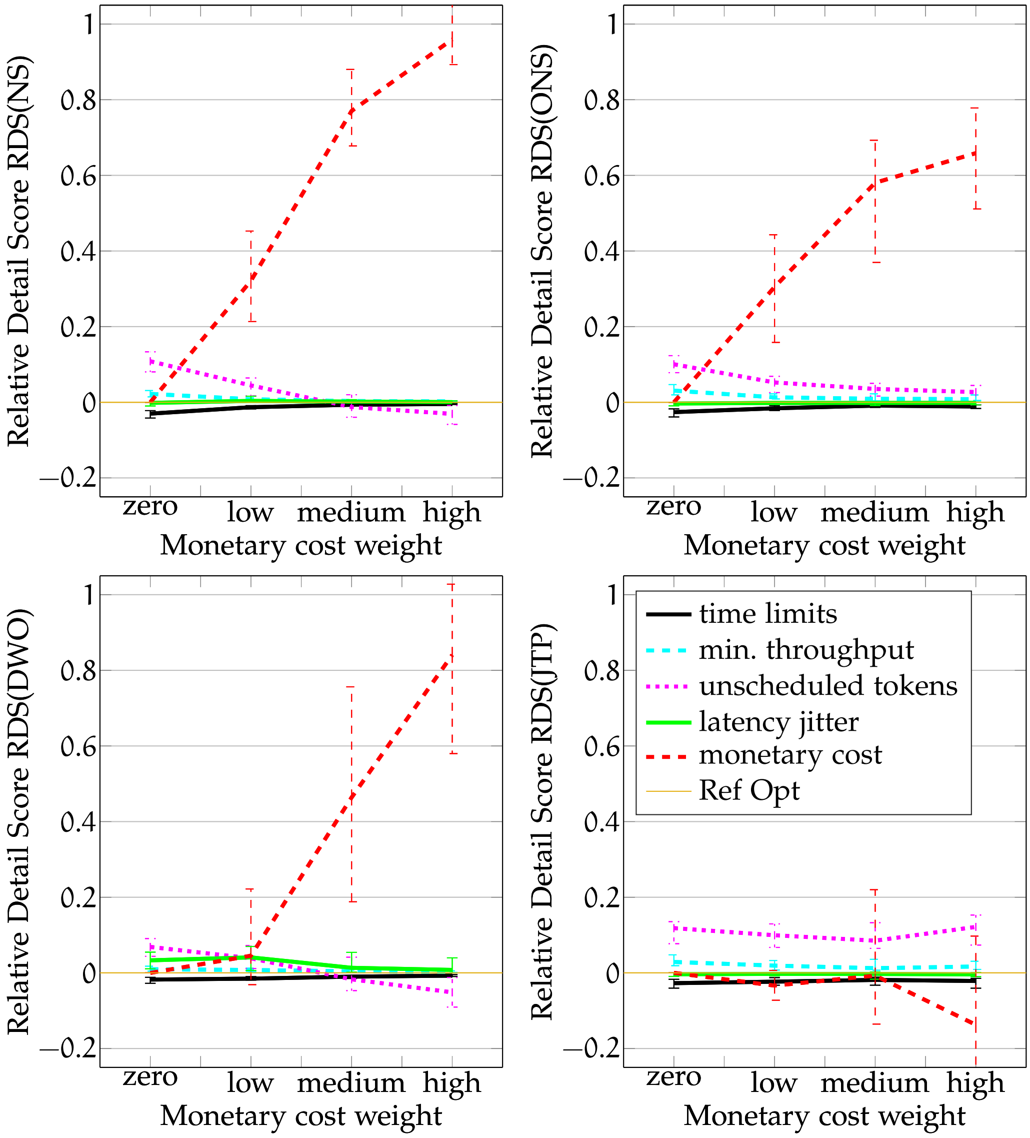


Figure 59: Relative Detail Scores of NS, ONS, DWO and JTP over monetary cost weight

A.5.4 *Adaptation Results with Relative Optimization Potential*

As it can be seen from Figure 60, the Relative Optimization Potential (ROP), illustrated as black dashed line, stays about constant with different error strengths. Joint Transmission Planning (JTP) using perfect prediction (green), i.e. not considering the error on the x-axis, and Opportunistic Network Selection (ONS) in dottet magenta frame the results of the Adaptation (solid black). A detailed discussion is given in 5.3.

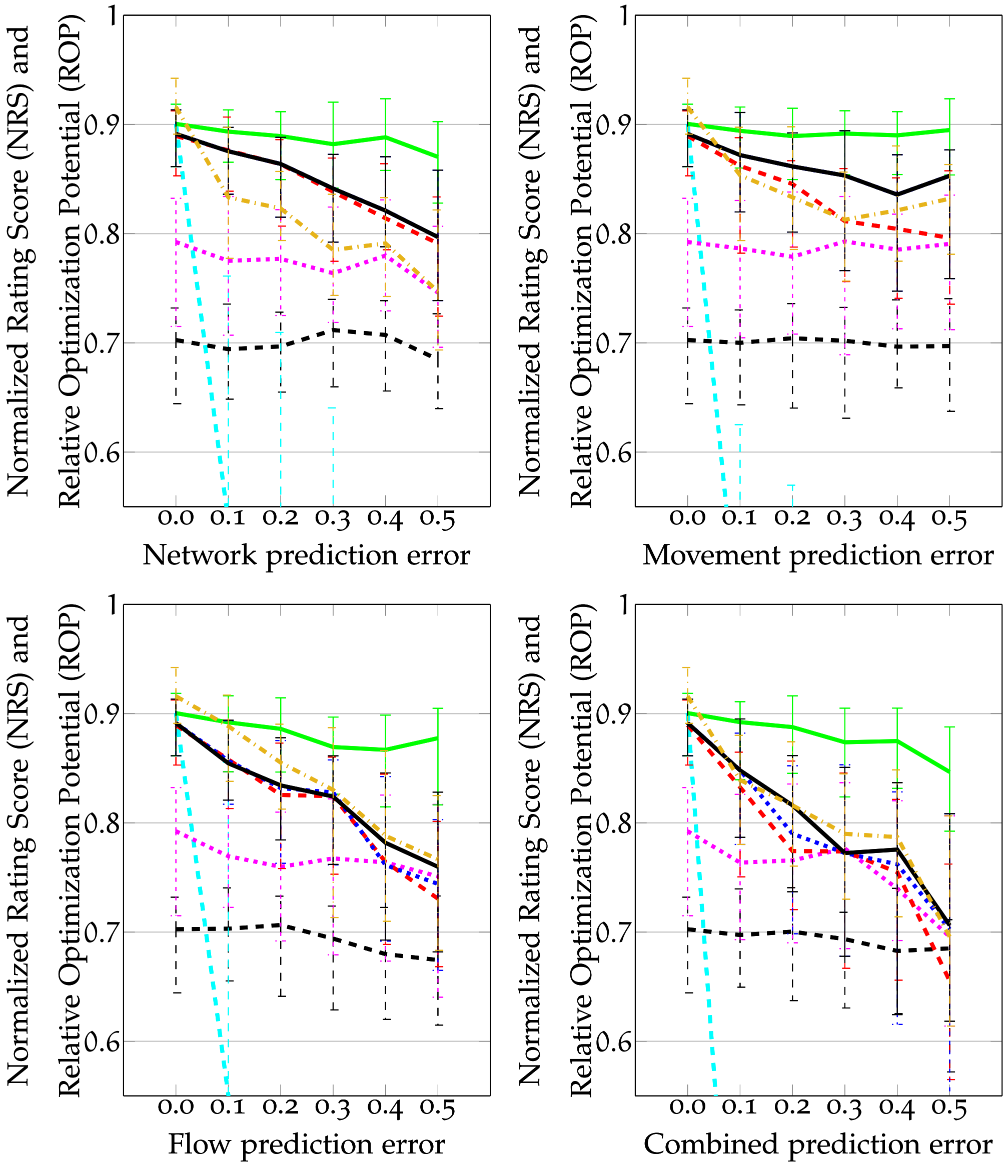
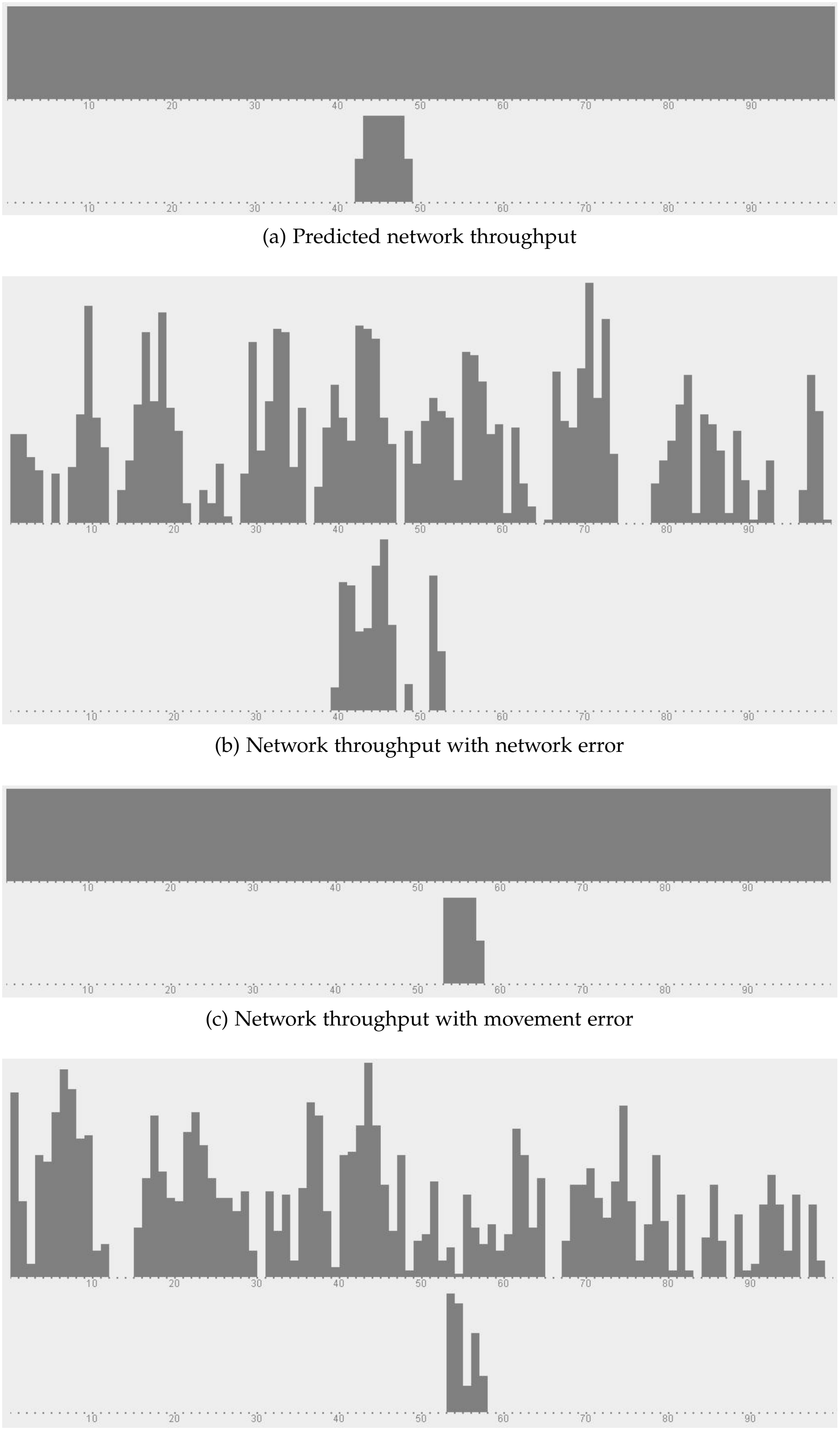


Figure 60: Planners' NRS over SMAPE: movement, network, data flow, combined and execution duration in seconds per instance with Relative Optimization Potential (ROP) as dashed black line

A.6 EXAMPLE OF APPLIED PREDICTION ERROR MODEL WITH SMAPE 0.5



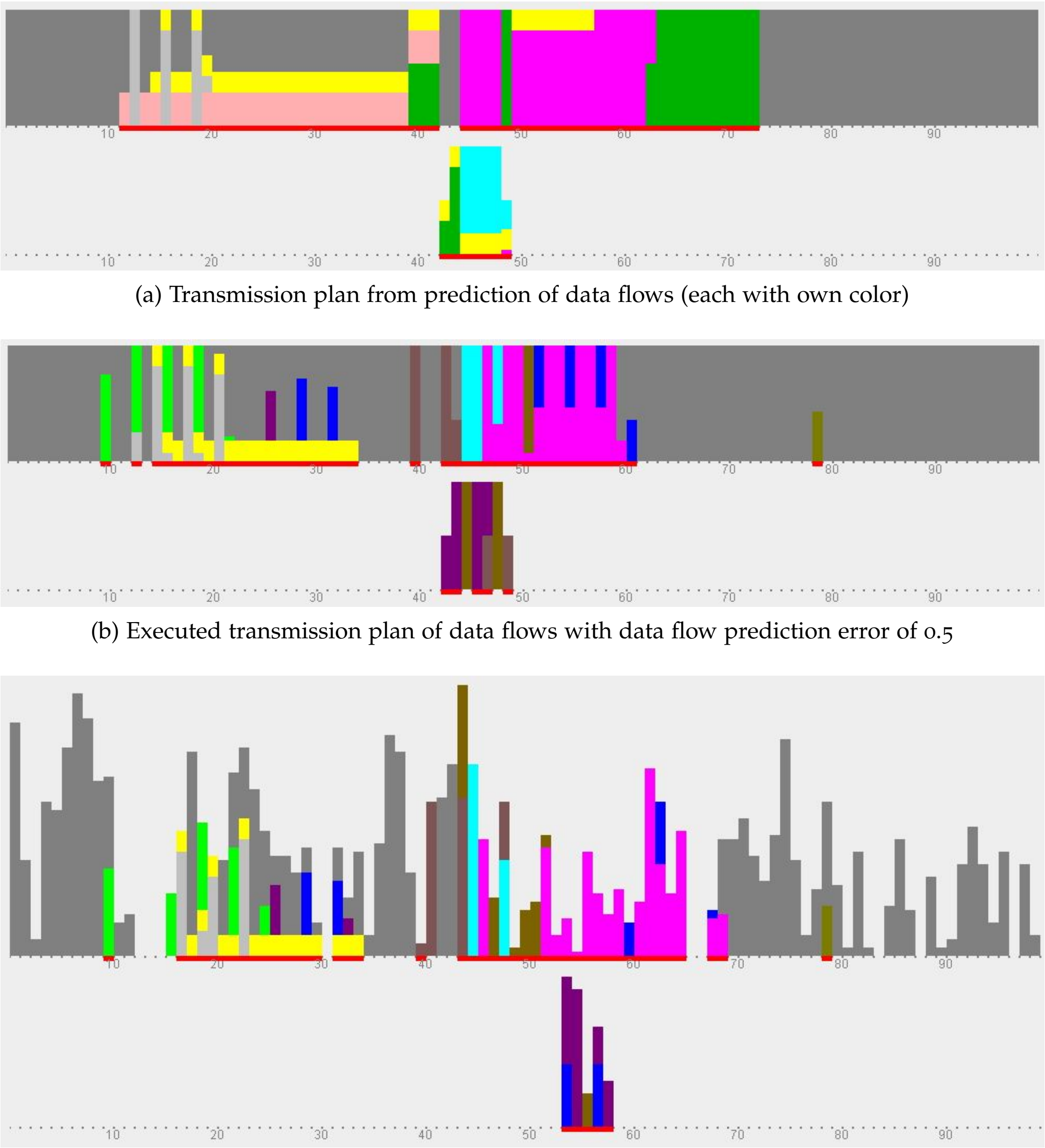
(a) Predicted network throughput

(b) Network throughput with network error

(c) Network throughput with movement error

(d) Network throughput with movement and network error

Figure 61: Throughput for two networks over time. Network and movement prediction error examples with SMAPE strength 0.5.



(a) Transmission plan from prediction of data flows (each with own color)

(b) Executed transmission plan of data flows with data flow prediction error of 0.5

(c) Executed transmission plan of data flows with combined prediction error of 0.5

Figure 62: Throughput for two networks over time. Network, movement and flow prediction error examples with SMAPE strength 0.5.

B PUBLICATIONS

B.1 MAIN PUBLICATIONS

1. T. Rueckelt, I. Stavrakakis, T. Meuser, I. H. Brahmi, D. Böhnstedt, and R. Steinmetz, *Data Transmission Plan Adaptation Complementing Strategic Time-Network Selection for Connected Vehicles*, in Elsevier Ad Hoc Networks (accepted for publication).

2. T. Rueckelt, D. Burgstahler, F. Jomrich, D. Böhnstedt, and R. Steinmetz, *Impact of Time in Network Selection for Mobile Nodes*, in Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 2016.

3. T. Rueckelt, I. Stavrakakis, T. Meuser, D. Böhnstedt, and R. Steinmetz, *Data Transmission Plan Adaptation for Connected Vehicles*, in Proceedings of the IEEE International Balkan Conference on Communications and Networking (BalkanCom), 2017.

4. T. Rueckelt, D. Burgstahler, D. Böhnstedt, and R. Steinmetz, *MoVeNet: Mobility Management for Vehicular Networking*, in Proceedings of the ACM MSWiM International Symposium on Mobility Management and Wireless Access (MobiWAC), 2016.

5. T. Rueckelt, D. Burgstahler, F. Englert, C. Gottron, S. Zöller, R. Steinmetz, *A Concept for Vehicle Internet Connectivity for Non-Safety Applications*, in Proceedings of the IEEE Local Computer Networks Conference (LCN), 2014.

6. T. Rueckelt, F. Jomrich, D. Burgstahler, D. Böhnstedt, and R. Steinmetz, *Publish-Subscribe-Based Control Mechanism for Scheduling Integration in Mobile IPv6*, in Proceedings of the IEEE Local Computer Networks Conference (LCN), 2015.

7. T. Rueckelt, C. Büttner, *Sicheres Service Management für ITS-Dienste*, in Proceedings of the Automotive meets Electronics (AmE), VDE Verlag GmbH, 2015.

8. F. Jomrich, T. Rueckelt, D. Böhnstedt, R. Steinmetz, *An Efficient Heat-Map-Based Wireless Communication Simulation Model for Omnet++*, in Proceedings of the Automotive meets Electronics (AmE), VDE Verlag GmbH, 2017.

9. T. Rückelt, M. Alhumaidi, A. Zoubir, *Realization of a Transverse Feedback System for SIS18/100 Using FPGA*, in Proceedings of the International Beam Instrumentation Conference (IBIC), 2013

10. T. Rückelt, GM Global Technology Operations LLC, *Verkehrssteuerung autonom fahrender Fahrzeuge*, Patent DE 10 2017 007 137, July 2017

B.2 CO-AUTHORED PUBLICATIONS

11. F. Jomrich, A. Sharma, T. Rueckelt, D. Burgstahler, D. Böhnstedt and, R. Steinmetz. *Dynamic Map Update Protocol for Highly Automated Driving Vehicles*, in Proceedings of the Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), SciTePress Science and Technology Publications Lda, 2017.

12. C. Büttner, T. Rueckelt, S. Huss, *Sicheres Hochladen, Austauschen und Verteilen von Daten in einem Car2X Systemverbund*, in Proceedings of the Automotive meets Electronics (AmE), VDE Verlag GmbH, 2016.

13. D. Burgstahler, F. Knapp, S. Zöller, T. Rueckelt and R. Steinmetz, *Where is That Car Parked? A Wireless Sensor Network-Based Approach to Detect Car Positions*, in Proceedings of the IEEE LCN International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp), 2014.

14. F. Jomrich, T. Wankhede, T. Rueckelt, D. Burgsthaler, D. Böhnstedt, R. Steinmetz, *Rapid Cellular Network Simulation Framework for Automotive Scenarios (RACE Framework)*, in Proceedings of the International Conference on Networked Systems (NetSys), 2017.

15. T. Meuser, D. Burgstahler, T. Rueckelt, D. Böhnstedt and R. Steinmetz *Hybrid-ProbSense.KOM: Probabilistic Sensing with Hybrid Communication for Gathering Vehicular Sensed Data*, in Proceedings of the Automotive meets Electronics (AmE), 2017.

16. D. Burgstahler, M. Pelzer, A. Lotz, F. Knapp, H. Pu, T. Rueckelt and R. Steinmetz, *A Concept for a C2X-based Crossroad Assistant*, in Proceedings of the IEEE PerCom Workshop on Smart Environments: Closing the Loop (SmartE), 2015.

17. D. Burgstahler, S. Zöller, M. Möbus, T. Walter, T. Rueckelt and R. Steinmetz, *Navigate.KOM: Datenbankbasierter Informationsansatz für Fahrassistenzsysteme*, in Proceedings of the Automotive meets Electronics (AmE), VDE Verlag GmbH, 2015.

C SUPERVISED STUDENT WORKS

C.1 MASTER THESES

1. Smart Cellular Network Probing Using Delay-Tolerant Data, Markus Grau (co-supervision), May 2017
2. In-situ Communication Schedule Adaptation for Vehicles, Dan Li, October 2016
3. Accurate Map Creation from Mobile Sensor Data, Pratik Kumar Mankar (co-supervision), September 2016
4. Design of a Mobility Protocol for Vehicular Networking, Halis Altug, October 2015
5. Data Traffic Scheduling in Heterogeneous Communication Networks, Jigar Modi, August 2015
6. Seamless Handover in Wireless Vehicular Communication Networks, Florian Jomrich, July 2015

C.2 BACHELOR THESES

1. Prototype Development of MoVeNet, Jan-Thomas Becker, April 2017
2. Artificial Intelligence-based Data Transmission Scheduling for Vehicular Communication, Jakob Weimar, October 2016
3. Data Flow Characteristics for Vehicle Applications: Analysis and Modeling, Jens Balze, December 2015

C.3 OTHERS

- How Can Data Transmission Profit from Time Selection?, Björn Büschke, Benedikt Böhning, KOM Seminar, February 2017
- Efficient Mobile Network Probing, Hourieh Hassan Hosseini, Internship (co-supervision), January 2017
- Analysis of TCP Behavior during Transparent Handover of MoVeNet, Jakob Weimar, KOM Lab, August 2016
- Schedule Assessment and Visualization, Dennis Hanslik, KOM Lab, February 2016

- Potential of Media Independent Handover (802.21) for Vehicular Communications, Chaitanya Venkatkrishna, Shubham Bhadwaj; Gaetan Kieffer, Steven Leduc, ATFIR Seminar March 2015
- Network Availability Prediction for Vehicular Communication, Sven Peldzus, KOM Lab, August 2014
- Wireshark plug-in and visualization for V2V, Amit Bhanja, Raghunath Deshpande, KOM Lab, August 2014
- Automotive Apps: ECOmfort, Martin Möbus, Tim Walter, KOM Project, February 2014
- Automotive Apps: Intelligent Refuel, Sven Peldzus, KOM Project, February 2014

D DANKE

Während meiner Promotion habe ich viel Unterstützung und Rückhalt von meinen Freunden, Kollegen und meiner Familie erhalten, denen ich an dieser Stelle herzlich danke. In der Zeit meiner Promotion war ich bei der Adam Opel AG beschäftigt. Hier bedanke ich mich besonders bei Harald Berninger, der mir auch neben der Promotion viele gute Erfahrungen mit auf den Weg gegeben hat. Weiterhin einen besonderen Dank an Carsten Büttner für viele spannende und unterhaltsame Diskussionen, auch abseits der Arbeit. Ich danke auch allen Kollegen im direkten Umfeld, insbesondere Thomas Streubel, Boliang Yi, Bernd Büchs, Steffen Knapp, Frank Bonarens und Stefan Berger, den Abteilungsleitern Nikolas Wagner und Bruno Praunsmändel und den weiteren Mitstreitern.

Mein Arbeitsumfeld war geprägt durch wöchentliche Wechsel zwischen Opel in Rüsselsheim und dem Multimedia Communications Lab (KOM) an der TU Darmstadt. Als Teil der Gruppe Distributed Sensing Systems bedanke ich mich für die fantastische Zusammenarbeit. Hier möchte ich Frank Englert und Sebastian Zöller besonders erwähnen, die mir in den ersten Monaten in vielen Diskussionen bei meiner Themenfindung geholfen haben, sowie bei den Gruppenleitern Christian Gottron, Sonja Bergsträßer, Doreen Böhnstedt und Björn Richterzhagen für die organisatorische und methodische Unterstützung. Für die gute Zusammenarbeit in der weiteren Entwicklung von Ideen, ihrer Ausgestaltung und Veröffentlichung bedanke ich mich besonders bei Daniel Burgstahler, Tobias Meuser, Florian Jomrich, An The Binh Nguyen, Alaa Ahalmoud und Patrick Lieser und allen weiteren KOMlern. Ich danke allen Studenten, die ihre Abschlussarbeiten und Seminare bei mir angefertigt haben und mit neuen Denkanstößen und viel Eifer zur Verwirklichung meiner Ideen beigetragen haben, insbesondere Halis Altug, Jan-Thomas Becker, Dan Li und Jakob Weimar.

Besonderer Dank gilt meinem Doktorvater Ralf Steinmetz, der mit dem Multimedia Communications Lab (KOM) einen hervorragenden Nährboden für die wissenschaftliche Arbeit auf höchstem Niveau unter Freunden geschaffen hat. Ich bin froh, ein Teil davon zu sein. Weiterhin danke ich meinem Korreferenten Ioannis Stavrakakis, der mich besonders in der Endphase meiner Arbeit unterstützt und zur Verbesserung meines Schreibstils beigetragen hat, sowie den weiteren Mitgliedern meiner Prüfungskommission Klaus Hofmann, Ulrich Konigorski und Franko Küppers. Besonders erwähnen möchte ich weiterhin Sorin Huss und Alexander Biedermann, die während meines Studiums meine ersten Schritte in Richtung Forschung begleiteten und den Spaß an der wissenschaftlichen Arbeitsweise in mir entflammten.

Nicht zuletzt danke ich meinen Freunden und meiner Familie für den stetigen Rückhalt und die bedingungslose Unterstützung, für motivierende Worte und Aufmunterungen in schweren Phasen und für gute Ablenkung, die mir oft half einen freien Kopf zu bekommen. Ich danke Kay Steffen, Yasmin Müller, Jonas Schönichen, Steffen Jäger, Nicolas Weber, Patrick Sauer, Christian Lotz, Janine Laudan und allen weiteren, mit denen ich schöne Tage und Erlebnisse teilte. Besonderer Dank gilt meinen Eltern Josef und Karin Rückelt, und meiner Schwester und Familie, Daniela, Mathias, Luz-Maria und Alexander Föller.

Ich wünsche mir, dass die vielen guten Kontakte und Freundschaften aus dieser Zeit auch in Zukunft bestehen bleiben und wir weiterhin viel Spaß zusammen haben. Danke!

E CURRICULUM VITAE

personal information

Name	Tobias Rückelt
Date of Birth	April 18, 1987
Place of Birth	Fulda, Germany
Nationality	German

professional experience

since 11/2017	Daimler AG Stuttgart, Germany Future Transportation Systems, Vans Future Transportation & Connectivity Solution Architect
8/2013 - 10/2017	Technische Universität Darmstadt Darmstadt, Germany Electrical Engineering & Information Technology Dept. Multimedia Communications Lab (KOM) Research Assistant
7/2013 - 6/2017	GM Europe Engineering, Adam Opel AG Rüsselsheim, Germany GME Electrical Systems, Infotainment & Electrification Advanced Technology Research Assistant

education

7/2011 - 10/2013	Technische Universität Darmstadt Darmstadt, Germany Information Systems Technology Master of Science
10/2007 - 7/2011	Technische Universität Darmstadt Darmstadt, Germany Information Systems Technology Bachelor of Science
8/1998 - 6/2007	Ulrich von Hutten Gymnasium Schlüchtern, Germany Abitur

ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe.

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 30. Oktober 2017

Tobias Rückelt, M.Sc.