



TECHNISCHE
UNIVERSITÄT
DARMSTADT

MECHANISM TRANSITIONS
in PUBLISH/SUBSCRIBE SYSTEMS

Adaptive Event Brokering for Location-based Mobile Social Applications

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

von

BJÖRN RICHERZHAGEN, M.SC.

Geboren am 6. September 1988 in Remscheid-Lennep

Vorsitz: Prof. Dr.-Ing. Anja Klein
Referent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Michael Zink

Tag der Einreichung: 9. Mai 2017
Tag der Disputation: 26. Juli 2017

Hochschulkennziffer D17
Darmstadt 2017

Dieses Dokument wird bereitgestellt von This document is provided by
tuprints, E-Publishing-Service der Technischen Universität Darmstadt.
<http://tuprints.ulb.tu-darmstadt.de>
tuprints@ulb.tu-darmstadt.de

Bitte zitieren Sie dieses Dokument als: Please cite this document as:
URN: [urn:nbn:de:tuda-tuprints-66692](https://nbn-resolving.org/urn:nbn:de:tuda-tuprints-66692)
URL: <http://tuprints.ulb.tu-darmstadt.de/id/eprint/6669>

Die Veröffentlichung steht unter folgender Creative Commons Lizenz:
Namensnennung - Keine Bearbeitungen 4.0 International
<https://creativecommons.org/licenses/by-nd/4.0/deed.de>

This publication is licensed under the following Creative Commons License:
Attribution-NoDerivatives 4.0 International
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>



ABSTRACT

INTERNET traffic caused by mobile devices will exceed the traffic originating from PCs by the year 2020, according to Cisco. With a projected 61-fold increase, location-based applications and augmented reality applications exhibit the fastest growth in the mobile sector. The success of interactive augmented reality games such as Google’s Ingress or Pokémon Go is the latest indicator for the increasing popularity of *location-based mobile social applications*. However, the direct interaction among users and the inherent locality of this interaction in such applications pose considerable challenges to a communication system. Efficient brokering of content based on the physical location of mobile clients is further complicated by application-specific attraction points and the resulting heterogeneities and dynamics of client mobility.

The publish/subscribe paradigm is well suited to model this interest-based communication pattern between users. Distinct mechanisms for both location-based filtering and locality-aware dissemination of events have been proposed for numerous application domains. However, their combined utilization and their adaptation to user mobility, network heterogeneity, and dynamic workload characteristics of mobile social applications remains an open research challenge.

In this thesis, we design and implement transitions between distinct mechanisms for location-based filtering and locality-aware dissemination of events as our *first contribution* to address this challenge. We develop a methodology for the encapsulation of mechanisms and the execution of transitions within our publish/subscribe framework BYPASS.KOM. Consequently, we apply our methodology to location-based filter schemes and dissemination mechanisms for locality-aware publish/subscribe, integrating state of the art mechanisms into BYPASS.KOM. We propose distinct execution strategies for transitions, focusing on the seamless operation of the publish/subscribe system during execution by means of state transfer between mechanisms.

By deriving common abstractions for transition-enabled mechanisms and the coordinated execution of transitions, we generalize our methodology as part of the SIMONSTRATOR.KOM platform. These abstractions for the design and evaluation of transition-enabled communication systems constitute our *second contribution*. This includes mobility and workload models for location-based mobile social applications.

Based on a prototype of BYPASS.KOM, we conduct an extensive evaluation of our contributions using the SIMONSTRATOR.KOM platform. We show that our state transfer mechanism and the proposed abstractions for transition-enabled mechanisms lead to a seamless execution of transitions. Additionally, we demonstrate the combined utilization of location-based filtering and locality-aware event dissemination as coexisting transition-enabled mechanisms. Overall, we show that our contributions allow the publish/subscribe system to adapt to application-specific mobility and workload characteristics by executing the respective mechanism transitions.

DAS von mobilen Endgeräten verursachte Datenaufkommen wird, laut einer Studie von Cisco, im Jahr 2020 den von PCs ausgehenden Anteil übertreffen. Das größte Wachstum verzeichnen dabei lokationsbasierte Dienste und so genannte *Augmented Reality* Anwendungen, deren Anteil laut dieser Studie im selben Zeitraum um das 61-fache steigen wird. Der Erfolg von Ingress und Pokémon Go als prominente Vertreter von Augmented Reality Anwendungen stützt diese Vorhersagen. Charakteristisch für solche mobilen sozialen Anwendungen ist die direkte Interaktion von Nutzern in räumlicher Nähe zueinander. Diese lokalen Interaktionsmuster und die interaktive Natur der Anwendungen stellen hohe Anforderungen an das zugrundeliegende Kommunikationssystem. Inhalte müssen basierend auf der aktuellen Position der Nutzer gefiltert und verteilt werden; ein Prozess, der zusätzlich durch den Einfluss der Anwendung auf das Verhalten der Nutzer erschwert wird. So führen Inhalte der Anwendung (beispielsweise die Entdeckung eines seltenen Pokémon) zu hoher Dynamik im Nutzerverhalten.

Das *Publish/Subscribe* Paradigma ist besonders gut geeignet um solche Interaktionsmuster auf ein Kommunikationssystem abzubilden. Entsprechende Mechanismen, sowohl für eine ortsbezogene Filterung von Inhalten, als auch eine umgebungsbewusste Verteilung der Inhalte an mobile Endgeräte, wurden bereits für andere Anwendungsbereiche vorgestellt. Deren gemeinsame Nutzung und entsprechende Techniken zur Anpassung an das dynamische Nutzerverhalten mobiler sozialer Anwendungen wurden bislang nicht erforscht.

In dieser Dissertation erforschen wir koordinierte Übergänge, sogenannte Transitionen, zwischen entsprechenden Mechanismen für ortsbezogene Filterung und umgebungsbewusste Verteilung von Inhalten in Publish/Subscribe Systemen. Unser erster Beitrag, *BYPASS.COM*, dient dabei als *Framework* zur Erforschung einer Methodik zur Kapselung von Mechanismen und der Ausführung entsprechender Transitionen zwischen diesen Mechanismen. Die resultierende Methodik wenden wir im Rahmen dieser Dissertation für bestehende ortsbezogene Filtermechanismen und umgebungsbewusste Verteilmechanismen an. Wir stellen entsprechende Strategien zur unterbrechungsfreien Durchführung von Transitionen vor, wobei wir insbesondere den Transfer von Zustandsinformationen während einer Transition ermöglichen.

Mit unserem zweiten Beitrag, der *SIMONSTRATOR.COM* Plattform, generalisieren wir unsere Methodik zur Kapselung von Mechanismen und der Durchführung von Transitionen. Wir stellen Modelle und Komponenten zur Verfügung, mittels derer transitionsfähige Mechanismen konzeptioniert, realisiert und evaluiert werden können. Dies beinhaltet Mobilitäts- und Netzmodelle für mobile soziale Anwendungen, welche für die Evaluation unseres Prototypen von *BYPASS.COM* verwendet werden.

Unsere Evaluation belegt, dass die von uns entwickelte Methodik zur Kapselung von Mechanismen und zur Durchführung von Transitionen die Anpassung eines

Publish/Subscribe Systems an anwendungsspezifische Dynamik ermöglicht. Dabei belegen wir insbesondere den Einfluss des Zustandstransfers und der von uns vorgeschlagenen Modelle und Komponenten der SIMONSTRATOR.KOM Plattform auf die unterbrechungsfreie Ausführung von Transitionen. Die Beiträge dieser Dissertation ermöglichen die gemeinsame und adaptive Nutzung von Mechanismen zur ortsbezogenen Filterung und umgebungsbewussten Verteilung von Inhalten in einem Publish/Subscribe System.

DANKSAGUNG

Alles muss man selber machen lassen singen Deichkind. Und auch mein Promotionsvorhaben hätte ich ohne die Unterstützung einer ganzen Reihe von *tremendous people* wohl niemals erfolgreich abgeschlossen.

Da wären zunächst einmal jene, die mir die Promotion überhaupt erst ermöglicht haben: Christian und Dominik, die mich als HiWi am Fachgebiet KOM betreut und mir so einen ungeschönten Einblick in das Leben eines gestressten Doktoranden gegeben haben. Julius und Matthias, die meine Masterarbeit betreut haben und mich damit direkt in den (damals geplanten) SFB MAKI integriert haben. Und natürlich Ralf Steinmetz, der mich 2013 in eben diesem Projekt angestellt hat.

Wo wir schon dabei sind: MAKI ist für mich der ideale Nährboden für gute Ideen – wo sonst findet man so viele Kollegen in nächster Nähe, die an einem gemeinsamen Problem arbeiten? Und wo sonst hat man dabei gleichzeitig so viele Freiheiten wie in MAKI? Ich bin stolz und dankbar, ein Teil dieses Teams zu sein. Ohne Anspruch auf Vollständigkeit möchte ich insbesondere Julius, Denny, Stefan, Michael, Jens, Marcel, Mousie, Alex, Roland, Matthias, Markus, Marc und natürlich Torsten danken!

Mehr noch als MAKI hat der Lehrstuhl selbst die vergangenen viereinhalb Jahre geprägt. KOM ist *kein* zusammengewürfelter Haufen von Einzelkämpfern. KOM ist vielmehr ein von Ralf Steinmetz gehegtes Gefüge aus Studenten, HiWis, ATMs, Promovierenden und Post-Docs, die weit über das jährliche Retreat im KWT hinaus miteinander verbunden sind. HiWis und Abschlussarbeiter haben einen wichtigen Beitrag zu meiner Dissertation geleistet – insbesondere Marc, beide Julians, Sophie, Thorben, Leo, Lukas, Dennis und der andere Björn. Euch allen vielen Dank für eure gute Arbeit, die ein oder andere Überstunde und eure kritischen Fragen!

Dann wären da die Kollegen: ist es Sarkasmus? Schwarzer Humor? Eins steht fest: Jeremias ist genau der richtige Büromitmieter für mich. Danke auch an die AOC-Kollegen der ersten Stunde: Rahul und Rhaban, der immer ein bisschen *unser Jüngster* bleiben wird. Und natürlich an beide Tobis von der anderen Ganghälfte und überhaupt meine zukünftigen Kollegen in der DSS-Gruppe: ich freue mich auf die nächste Zeit mit euch! Herausstellen möchte ich auch die Kollegen, die immer mal wieder die Treppe zwischen erstem und zweitem Stock bei KOM überwinden und so für gruppenübergreifende Aktivitäten sorgen: Steffen und Thomas, da seid natürlich besonders ihr gemeint. Und: Frank und Daniel – euch hätte ich ja schon gerne noch ein bisschen länger am Lehrstuhl gesehen...

Dann ist da noch das, *was die Welt im Innersten zusammenhält*: unsere ATMs natürlich. Britta, Moni, Sabine, Karola, Frank, Jan, Frau Scholz-Schmidt und Frau Ehlhardt – zufälligerweise auch Gründungsmitglieder des *DJ Börnie Fanclubs*. Bei euch kann man sich gar nicht oft genug bedanken: für die vielen, vielen kleinen und großen Dinge im Lehrstuhllalltag, die ihr mir abgenommen und erleichtert habt. Aber eben auch: für die guten Gespräche, Ratschläge und schönen Momente im KWT, auf dem

Heinerfest und auch mal in der Kaffeeküche. Ihr seid das Rückgrat des Lehrstuhls und damit auch jeder einzelnen Promotion. In diesem Zusammenhang natürlich auch ein großes Dankeschön an Ralf Steinmetz, der das alles hier überhaupt erst ermöglicht und zusammenhält.

In so einer Umgebung werden aus Kollegen Freunde.¹ Aus Dominik wurde Stingl (zu ganz besonderen Anlässen auch mal Domi), aus Paul wurde Baumann, Wendel wurde zu Wendler, und Hans – Hans bleibt Hans. Dieses Schicksal teilen auch Saller und Lieser. Und bei euch möchte ich mich hiermit ganz herzlich bedanken: für gemeinsame Skiurlaube, Festival-Besuche, Konzert- und Stadiongänge, Havana-Brunches und – natürlich – das ein oder andere Feierabendbier. Aber auch für spannende Diskussionen, hilfreiches Feedback, generellen Rat in allen Lebenslagen und überhaupt eine Menge Unterstützung, wenn es mal wieder ein bisschen viel wurde mit Deadlines, Papern, Projekten und Dissertation. Danke! Und: bitte weiter so!

Auf den letzten Metern haben mir insbesondere meine beiden Leidensgenossen Ronny und Fabian geholfen – sowohl durch gegenseitiges Aufbauen als auch durch die Möglichkeit, ein bisschen Frust loszuwerden, wenn es mal wieder nicht so wirklich mit dem Schreiben voran ging. Großer Dank gebührt auch meinem Korreferenten Michael Zink, der mir neben inhaltlichem Feedback zur Dissertation auch immer die nötige Portion Gelassenheit vermittelt hat und trotz Umzugstress einen kühlen Kopf in der Schlussphase bewahrt hat. Vielen Dank auch an meinen Gruppenleiter Boris, der sich trotz knappem Zeitplan immer wieder durch seitenweise Text gewühlt hat und mich dann doch dazu getrieben hat, noch die ein oder andere hilfreiche Abbildung einzubauen.

Die wichtigsten Menschen kommen natürlich ganz zum Schluss. Ich habe das große Glück, mit Nils nicht nur den *besten kleinen Bruder der Welt*, sondern auch einen unglaublich talentierten und gewissenhaften Kollegen am Lehrstuhl zu haben. Für die ein oder andere nächtliche Paper-Session, viele tolle inhaltliche Gespräche und generell für die Gewissheit, ein Stück Familie immer in greifbarer Nähe zu haben: Dankeschön! Ich wünsche dir für deine Promotion viel Erfolg – aber Sorgen mache ich mir da nicht. Neben der Zweigstelle in Darmstadt gibt es natürlich noch den Familienhauptsitz in Lennep; hier halten meine Eltern die Stellung. Euch beiden kann ich gar nicht wirklich sagen, wie dankbar ich für die Unterstützung in jeder Lebenslage bin. Egal ob Schule, Studium oder jetzt die Promotion: ihr habt jeden Schlenker meinerseits nicht nur hingenommen, sondern mich auch immer wieder ermutigt. Auf eure Unterstützung in jeder Hinsicht konnte und kann ich mich immer verlassen. Vielen Dank dafür!

Und *last, but not least*: vielen lieben Dank an dich, Sophie, für die unendliche Geduld mit mir während der teilweise doch etwas längeren Durststrecken auf dem Weg zur Einreichung. Ich kann dir nicht genug danken für die (mindestens) 1703 kleinen Aufmunterungen und Aufmerksamkeiten in den letzten Monaten. Ich liebe dich.

Danke!

¹ Was natürlich nicht heißen soll, dass alle hier nicht genannten aber vorher erwähnten Personen für mich *nur* Kollegen sind...

CONTENTS

1	INTRODUCTION	1
1.1	Motivation for Adaptive Event Brokering	2
1.2	Research Challenges	3
1.3	Research Goals and Contributions	4
1.4	Structure of the Thesis	5
2	BACKGROUND	7
2.1	Location-based Mobile Social Applications	7
2.1.1	Mobile Augmented Reality Games	9
2.1.2	Interaction and Attraction Characteristics	10
2.2	Event Brokering and the Publish/Subscribe Paradigm	11
2.3	Communication Networks	13
2.4	Concept of Mechanism Transitions	17
3	STATE OF THE ART	21
3.1	Mobility Support in Publish/Subscribe Systems	21
3.1.1	Producer and Consumer Mobility	21
3.1.2	Broker Mobility in Mobile Ad Hoc Networks	24
3.1.3	Interest Mobility through Location-based Filtering	26
3.2	Adaptivity in Publish/Subscribe Systems	29
3.2.1	Broker-centric: Adaptivity in Distributed Broker Networks	29
3.2.2	Client-centric: Adaptivity through Reconfigurable Middleware	30
3.3	Adaptivity through Mechanism Transitions	31
3.4	Summary and Identified Research Gap	33
4	BYPASS.KOM: TRANSITIONS IN EVENT BROKERING	35
4.1	Conceptual Overview	35
4.2	Transitions between Location-based Filter Schemes	37
4.2.1	Model and API for Location-based Subscriptions	38
4.2.2	Encapsulation of Filter Schemes	42
4.2.3	Executing Transitions between Filter Schemes	46
4.2.4	Integration of Existing Mechanisms	49
4.2.5	Mechanism Reconfiguration through Self-Transitions	55
4.3	Transitions between Event Dissemination Mechanisms	56
4.3.1	Local Ad Hoc Event Dissemination	57
4.3.2	Cloud-based Hybrid Event Dissemination	60
4.3.3	Executing Transitions between Local Dissemination Mechanisms	61
4.3.4	Locality-aware Dissemination with Stateless Gateways	63
4.4	Coexistence of Transition-enabled Mechanisms	65

5	SIMONSTRATOR.KOM: PLATFORM FOR MECHANISM TRANSITIONS	69
5.1	Overview of the Platform	69
5.1.1	The Core Framework: Time, Events, and Instrumentation	70
5.1.2	Component-based Composition of Hosts	72
5.1.3	Runtime Environments	73
5.2	Platform Support for Transitions	73
5.2.1	Transition-enabled Mechanisms and Mechanism Proxies	74
5.2.2	Transition Engine and Custom Transitions	76
5.2.3	Coordination of Transitions on Multiple Hosts	79
5.2.4	Monitoring, Analysis, and Planning	79
5.3	Prototype of Bypass.KOM	81
5.3.1	Mechanism Proxies Utilized by Clients and Brokers	81
5.3.2	Coordination of Transitions for Coexisting Mechanisms	83
6	EVALUATION OF MECHANISM TRANSITIONS	89
6.1	Evaluation Setup	90
6.1.1	Mobility Models	90
6.1.2	Workload Models	94
6.1.3	Network Model	96
6.1.4	Evaluation Metrics	97
6.2	Transitions between Filter Schemes	99
6.2.1	Centrally Coordinated Total Transitions	100
6.2.2	Impact of the State Transfer Mechanism	101
6.2.3	Self-Transitions for Filter Scheme Adaptation	104
6.2.4	Partial Transitions vs. Self-Adapting Mechanisms	107
6.3	Transitions between Event Dissemination Mechanisms	111
6.3.1	Pure Ad Hoc Event Dissemination	111
6.3.2	Hybrid Dissemination Mechanisms	114
6.3.3	Globally Coordinated Transitions	118
6.3.4	Locally Initiated Transitions with Self-Healing	120
6.4	Coexistence of Transition-enabled Mechanisms	124
6.4.1	Direct Dissemination of Local Events near Attraction Points	124
6.4.2	Utilizing Gateways nearby Attraction Points	126
6.4.3	Executing Mechanism Transitions based on Client Density	129
7	SUMMARY, CONCLUSIONS, AND OUTLOOK	135
7.1	Summary of the Thesis	135
7.1.1	Contributions	135
7.1.2	Conclusions	136
7.2	Outlook	137
	BIBLIOGRAPHY	139
A	APPENDIX	157
A.1	Extended Study of Filter Schemes	157

A.2	Partial Transitions between Filter Schemes	163
A.3	Extended Study of Dissemination Mechanisms	165
A.4	List of Acronyms	169
A.5	Supervised Student Theses	171
B	AUTHOR'S PUBLICATIONS	173
C	CURRICULUM VITÆ	177
D	ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG	179

INTRODUCTION

THE expansion of mobile broadband networks and the advance of smartphones into our every-day life have changed the way we interact with communication technology. With 3.7 billion mobile broadband subscriptions in 2016 [39], ubiquitous access to the Internet is taken for granted. By 2020, about 30% of all Internet traffic is predicted to be caused by smartphones according to Cisco [25], for the first time exceeding the amount of traffic originating from PCs. The highest growth with approximately 24% each year until 2020 is predicted for *location-based services* [26].

From
location-based
services ...

Location-based services exploit the current location of the mobile device to determine the relevance of information for a user [165]. Prominent examples are navigation applications like Google Maps, recommender portals like Yelp or TripAdvisor (*top five restaurants nearby?*), and location-based advertising [34, 203].

... and mobile
social apps ...

At the same time, there is a shift towards *mobile social applications*, comparable to the success of online social networks. A mobile social application enables and fosters direct interaction between users. Such interaction already follows coarse location-dependent patterns in basic messaging applications such as *WhatsApp* or *Telegram* [164, 196]. However, locality in the interaction between users becomes even more significant for mobile social applications that explicitly incorporate location-based services.

Examples for the resulting *location-based mobile social applications* constitute communities for sports and hobbies (e.g., *Sporty*, *PlayWith*, or *bvddy*), location-based messaging applications (e.g., *Lokin*, *happn*) and mobile augmented reality games such as Google's *Ingress* or the recently introduced *Pokémon Go*. In augmented reality games, the physical world surrounding a user is augmented with game-specific information and interaction possibilities. Thus, users need to move around in the physical world and interact with real-world objects and other users in their proximity [38]. In 2015, the traffic caused by augmented reality applications increased by a factor of four compared to 2014 [26]. Based on the tremendous success of *Ingress* and *Pokémon Go*, the 61-fold increase predicted by Cisco until 2020 seems reasonable.

... to
augmented
reality games.

This traffic growth and the increasing significance of direct interaction among users in location-based mobile social applications pose major challenges to a communication system. Efficient dissemination of content created and consumed by mobile users requires mechanisms for location-based filtering and, consequently, locality-aware dissemination. These mechanisms need to adapt to the inherent dynamics in terms of mobility and network heterogeneity of mobile social applications.

Challenges for
communica-
tion
systems

In this thesis, we propose a methodology for and, consequently, a realization of, transitions between publish/subscribe mechanisms to address the aforementioned challenges. In the following, we motivate our approach to adaptive event brokering for location-based mobile social applications.

1.1 MOTIVATION FOR ADAPTIVE EVENT BROKERING

The *publish/subscribe* paradigm [40] is particularly well suited to model direct interaction between users and the interest-based communication pattern of mobile social applications. Users specify their interest in certain contents (hereafter referred to as *events*) through *subscriptions* that are issued to a *broker*. The broker (or, for scalability reasons, a network of brokers) filters incoming events based on these subscriptions and determines the set of users to which the events need to be disseminated. In this thesis, we focus on the adaptive combination of location-based filtering and locality-aware dissemination of events as core components of event brokering.

Brief history
of publish/sub-
scribe

Early research on publish/subscribe systems focused on the organization of distributed broker networks to increase scalability or to account for potentially unreliable brokers [17, 105, 125, 135]. Techniques for efficient bundling [183] and merging of similar subscriptions [53, 124, 133] were proposed to reduce the load caused by forwarding events and subscriptions in the broker network. While these approaches initially focused on static clients, broker networks were later extended to support mobile publishers and subscribers. Here, research focused on dealing with connection failures [15, 126] and the migration of client state between brokers [22, 24, 43, 51, 70, 132]. Bandwidth limitations of the cellular connection were addressed by aggregating events before sending them to mobile clients [71, 166] or even by discarding certain events [126]. While the aforementioned approaches support physical mobility of clients, they do not enable location-based filtering of events, i. e., logical mobility.

Location-
based
filtering

Support for location-based filtering of events can either be added by extending existing publish/subscribe systems [13, 30, 43, 51, 78, 202] or by introducing new subscription models for location-based filters [6, 11, 20, 41, 108, 200]. By extending existing systems, approaches can benefit from optimizations within channel- or attribute-based publish/subscribe systems [40]. However, they lack support for frequent location updates or fine-grained location-based subscriptions [30, 43]. Therefore, other approaches introduce new subscription models to offer location-based filtering as a native operation in the publish/subscribe system. The resulting dynamic nature of subscriptions poses a challenge to optimizations for broker networks [16], such as subscription merging [53, 133] and interest clustering [79, 102, 104].

Locality
awareness

While the aforementioned approaches support location-based filtering, they do not exploit the locality properties within the broker network. Locality of interaction is a key property of recent location-based mobile social applications such as the aforementioned augmented reality games, where user interaction is mostly determined by physical proximity. Several mechanisms exist to support locality-aware event dissemination, comprising *i*) offloading mechanisms [138, 142], where predetermined gateways communicate with the broker on behalf of other users, and *ii*) fully decentralized brokering within ad hoc networks formed by mobile users [45, 72, 120, 121]. By exploiting locality and mobile devices' distinct communication interfaces, these mechanisms reduce the load on the cellular infrastructure.

The efficiency of the aforementioned mechanisms depends on environmental conditions, such as the number of users, their mobility characteristics, and the avail-

ability of additional infrastructure (e. g., Wi-Fi access points). The dynamics of these conditions and the frequency of changes varies substantially for location-based mobile social applications [8]. Thus, the underlying publish/subscribe system needs to adapt accordingly. This is currently achieved by *i*) reconfiguring the mechanism itself [54, 66, 67] or *ii*) switching to another mechanism in a middleware [31, 168].

*Need for
adaptivity*

While the former can be achieved at low cost and high frequency, the range of conditions covered by reconfiguring a single mechanism is limited [8]. The latter approach can cover a wider range of environmental conditions; however, exchanging mechanisms is a costly and slow operation in current publish/subscribe middleware platforms, preventing its application in highly dynamic scenarios [31]. Still, according to [8], systems need to “[...] self-adapt autonomously by dynamically combining most suitable data distribution methods and techniques [...]”. Recently, coordinated mechanism transitions [46, 141, 192] beyond the limitations of middleware platforms have proven to enable systems to adapt to a wide range of conditions at low cost [157, 161, 191]. Therefore, *i*) composing mechanisms for location-based filtering and locality-aware event dissemination and *ii*) realizing transitions between different combinations of these mechanisms constitute essential steps towards adaptive event brokering for mobile social applications, as addressed in this thesis.

*Mechanism
transitions*

1.2 RESEARCH CHALLENGES

Mobile social applications impose challenges on mechanisms for location-based filtering and locality-aware dissemination of events. The following research challenges affect our proposed methodology for transitions between these mechanisms and the realization of the respective transitions in a publish/subscribe system.

Challenge: *Adaptation to user mobility and interaction patterns*

In location-based mobile social applications, user mobility and interaction with the application are closely intertwined. Application-specific interaction patterns, e. g., with points of interest, affect the efficiency of individual publish/subscribe mechanisms due to the resulting heterogeneity in terms of workload and client mobility. Addressing the resulting workload and mobility characteristics in a publish/subscribe system pose significant challenges given the limitations of individual mechanisms for location-based filtering and locality-aware dissemination of events. Consequently, our methodology for transitions between distinct mechanisms is required to adapt the publish/subscribe system to these application-specific characteristics.

Challenge: *Efficient utilization of heterogeneous networks*

Cellular plans are limited in terms of bandwidth and traffic volume, requiring efficient utilization of available resources to avoid decreasing user satisfaction [55]. Consequently, locality-aware event dissemination mechanisms can utilize Mobile Ad Hoc Networks (MANETs) to offload traffic from the cellular network, relying on distinct communication interfaces available on today’s mobile devices. Here, communication characteristics change rapidly as a consequence of user mobility. Therefore,

the respective mechanisms need to deal with sudden degradations in bandwidth and even complete loss of connectivity between devices. Additionally, the capacity of a MANET is limited, requiring careful consideration of how and when ad hoc communication is utilized. Finding a suitable combination of centralized (cellular) communication and decentralized (ad hoc) communication and adapting the choice of mechanisms to address mobility while still offloading the cellular network pose significant challenges. This includes the selection of suitable gateways, i. e., users that receive an event via the cellular connection and forward it in an ad hoc fashion.

1.3 RESEARCH GOALS AND CONTRIBUTIONS

The main goal of this work is the development of *a methodology for transitions between publish/subscribe mechanisms* to provide adaptive event brokering for location-based mobile social applications, and, consequently, *their realization and evaluation*. This objective is divided into the following three major research goals.

Research Goal 1: *Identification of mechanisms and potential transitions.*

To identify promising mechanisms and potential candidates for transitions, a classification of existing approaches to adaptive event brokering needs to be conducted. We focus on two aspects: *i)* mechanisms for locality-aware event dissemination and location-based filtering, as well as *ii)* adaptation strategies and their realizations within the communication system. The former is required to derive suitable abstractions for *transition-enabled mechanisms* within a publish/subscribe system. The latter aids in defining requirements and strategies for the *seamless execution* of transitions.

Research Goal 2: *Realization of seamless mechanism transitions.*

To realize adaptive event brokering, we require a methodology for the encapsulation of mechanisms and the execution of transitions. Our focus lies on the seamless execution of transitions between individual mechanisms in a publish/subscribe system. By executing transitions, the system can benefit from the mechanism that is most suitable for the current environmental conditions. Given the dynamics of the mobile social applications, the overhead of a transition execution and the impact on system performance during the execution needs to be carefully considered. Consequently, for a seamless transition, state information gathered by an active mechanism needs to be utilized during the transition to another mechanism. In our contributions, this problem is tackled from two perspectives: *i)* mechanism-specific within the domain of publish/subscribe [142, 144, 146, 147, 149] and *ii)* generalized to ensure extensibility with new mechanisms and applicability to other domains [49, 148, 155, 191].

Research Goal 3: *Evaluation and characterization of mechanism transitions.*

The effects of transitions between different mechanisms within the envisioned adaptive publish/subscribe system are to be measured. To this end, we contribute models for relevant features of the scenario of location-based mobile social applications, in-

cluding mobility and workload models [145, 149, 155, 186]. To characterize the transition itself, we focus on the behavior of the system right during as well as after the execution of a transition. The evaluation should furthermore prove the applicability of our proposed generic design concepts for transition-enabled systems [49].

With the thesis focusing on mobile applications, we explicitly concentrate on the communication between a broker and mobile clients and on direct communication among mobile clients. We strive to support existing approaches for distributed broker networks within the proposed system design to benefit from the rich body of existing research. Consequently, we do not propose novel distributed broker networks.

When mobile users need to contribute own resources, e. g., for direct communication, incentives and protection against malicious clients are relevant research topics. While we pinpoint to relevant related work, we do not conduct own work in this area. The same holds true for privacy concerns regarding location-based services.

In this work, transitions are triggered reactively based on the currently observed state of the system. Combining model-based predictions of the future system behavior with additional external data sources is expected to further increase the efficiency of transitions. Realizing proactive planning of transitions is not a part of this thesis, but clearly constitutes a promising direction for future research.

*Focus on
mobile clients*

*Incentives,
privacy, and
security*

*Reactive
vs. proactive
transitions*

1.4 STRUCTURE OF THE THESIS

Extending this brief introduction, we provide additional background on location-based mobile social applications, communication systems, event brokering, and mechanism transitions in Chapter 2. We discuss and classify related work in Chapter 3, addressing mobility support and means for adaptivity in publish/subscribe, further including a brief survey of mechanism transitions in other application domains.

Based on our discussion of related work, we propose *BYPASS.KOM* in Chapter 4. *BYPASS.KOM* is a modular framework allowing us to design, realize, and evaluate mechanism transitions in publish/subscribe systems. We discuss individual mechanisms, the transitions realized in the framework, and their execution strategies.

Our findings are generalized in Chapter 5, where we introduce the *SIMONSTRATOR.KOM* platform as an environment for the design and evaluation of mechanism transitions. We introduce core concepts of the platform and detail our transition-specific contributions before presenting our prototype realization of *BYPASS.KOM*.

We conduct an extensive evaluation of *BYPASS.KOM* and its mechanisms for location-based filtering and locality-aware event dissemination in Chapter 6. There, we specifically focus on the seamless execution of transitions by means of state transfer and different coordination concepts, showing detailed performance and cost characteristic for mechanism transitions. Additionally, we assess the general applicability of our methodology and design concepts for transition-enabled systems.

The thesis is concluded in Chapter 7 with a brief summary of the core contributions. Finally, we provide an outlook on potential future work.

BACKGROUND

IN the following, we provide relevant background information on the scenario of location-based mobile social applications as motivated in Chapter 1. We start by discussing the respective applications, involved entities, and their interaction patterns in Section 2.1. The interaction patterns and entities of a mobile social application are modeled with the publish/subscribe communication paradigm, which is introduced in Section 2.2. This discussion is followed by a brief primer on relevant technical aspects of the underlying communication networks in Section 2.3. Lastly, we detail the concept of mechanism transitions in the context of communication systems, providing the foundation for the discussion of the state of the art in Chapter 3.

2.1 LOCATION-BASED MOBILE SOCIAL APPLICATIONS

As introduced in Chapter 1, location-based mobile social applications combine the characteristics of *location-based services* and *mobile applications*, with a focus on direct *social interaction* between users [165]. Before going into detail on the aspect of social interaction, we briefly summarize characteristics of mobile applications in general.

A mobile application is designed for a specific mobile Operating System (OS), usually through a dedicated *app store*. Examples for such stores include the Google Play Store or Apple's App Store for the Android and iOS Operating System, respectively. Being built against the programming interfaces and libraries of the respective OS, applications can benefit from a range of platform-specific features. Most notably, mobile applications can request access to the mobile device's sensors and contextual information. All major platforms provide specific programming interfaces allowing applications to request the current location of a device, information about its connectivity, and even higher-level information about a user's current activities.

*Mobile
applications*

A large fraction of the more traditional location-based applications simply provides user-friendly access to a geographical database or directory service. Examples include the search for nearby amenities like restaurants, ATMs, or bus stops [97]. The raw location data is further augmented with application-specific information, such as reviews of restaurants or the next departures at a nearby bus stop. Most notably, mobile users do not interact with each other via the application, but instead consume information from the central, cloud-based service back end. The back end usually utilizes a geographical database [21, 75] to store and retrieve content that is relevant at, or around, a specific location.

*Location and
additional
context*

Information required at the service back end can either be generated by other users of the application (as with crowd-sourced restaurant reviews) or it can be fetched from external data sources and services (e. g., departure times). The number and type of utilized data sources heavily depends on the purpose and scope of the application.

Consequently, more general purpose mapping applications such as Google Maps or Tripadvisor utilize and aggregate a large set of data sources.

In addition to the user's current location, context information such as upcoming calendar entries or current weather data is utilized to filter what is presented to the user even further. As most users keep their smartphone within arm's reach throughout the day, location-based applications have a great potential for personalization, especially if they fuse large amounts of data and context sources [197]. This becomes apparent from the recent trend towards personal assistants such as Apple's Siri or Google Now. Rather than solely reacting to users' input by fetching data from the cloud-based back end, these applications deploy a rather push-based data model. Here, information is gathered by the back end and actively presented to the user, if a set of contextual conditions is met. A simple example are location-based reminders, as provided by all the aforementioned applications: Here, a user can set a reminder not only at a given date and time, but at a specific location (e. g., a supermarket). As soon as the user approaches said location, the application notifies the user.

In addition to geographical databases and a centralized back end, distributed approaches for location-based search and retrieval of information have been proposed [57, 60, 96], based on the concept of Peer-to-Peer (P2P) search overlays [56]. Here, information is stored in a distributed fashion involving either user PCs in a fixed network [57, 58] or mobile devices in an ad hoc network [59].

*Social
interaction
and local
communities*

Recently, location-based applications become increasingly focused on social interaction among users. Examples mentioned in the introduction are communities for sports and hobbies (e.g., *Sporty*, *PlayWith*, or *bvddy*) and location-based messaging applications (e.g., *Lokin*, *happn*), just to name a few. Here, the interaction among users exhibits certain locality characteristics, as verified in several user studies [52, 164, 196]. Users within close proximity tend to communicate with each other more often, depending on the target audience of an application. *Lokin*, for example, is an application provided by the Deutsche Bahn that enables communication with other users when commuting. It offers communication with other users on the same train ride, thereby exhibiting strong locality properties.

*Locality of
content and
interest*

We refer to this effect as *locality of content and interest*. The importance of addressing interest locality has also been studied in the context of learning communities [139]. Here, Rensing et al. argue that the context and, most importantly, the location of users needs to be taken into account for both learners and content authors. The same effect is studied in [207, 208] for video streaming behavior in a campus network. Through an extensive measurement study, the authors confirm our intuitive understanding that video clips showing locally relevant content (e. g., recorded within proximity) exhibit higher local popularity among users. Locality properties as observed in these applications become even more significant for mobile augmented reality games [101], as discussed in the following section.

2.1.1 Mobile Augmented Reality Games

A prominent example for location-based mobile social applications are mobile augmented reality games. They exhibit characteristics of location-based applications, while at the same time focusing on live interaction between nearby users. In the following, we briefly discuss the core characteristics of such applications, as they have a significant impact on the utilization of the underlying communication system.

As the name suggests, an augmented reality game embeds its game-specific content into our physical world, rather than presenting a fully virtual environment as done in virtual reality applications and games. In a virtual reality application, a head-mounted appliance replaces a user's visual perception of the physical surroundings with the content generated by the respective application. The virtual reality application reacts to movements of the user's head and to additional sensor input, often collected by hand-held controllers. While this setup achieves a very immersive user experience through the manipulation of visual perception and additional use of audio, it also confines the actual physical mobility.

*Augmented
vs. virtual
reality*

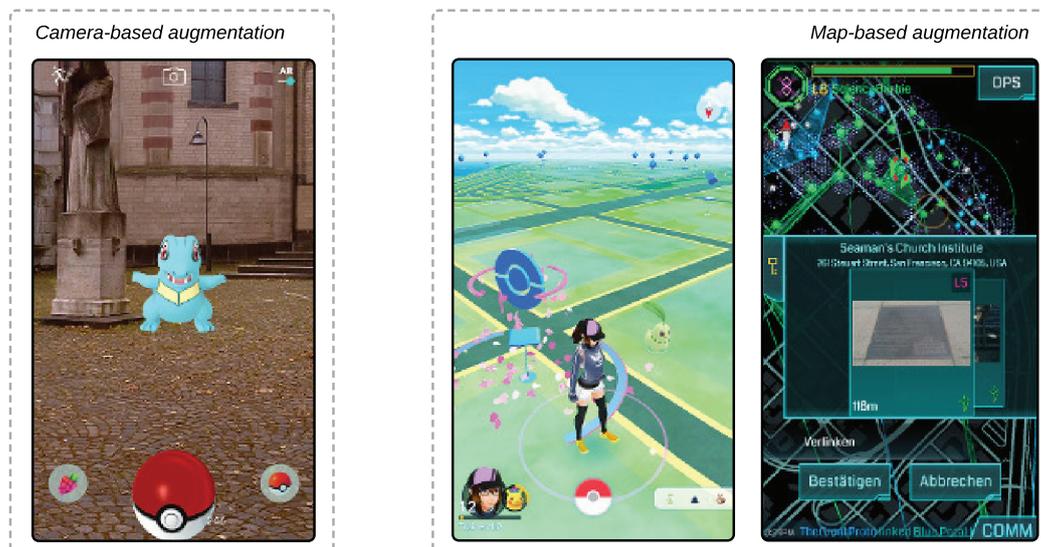


Figure 1: Examples for camera-based and map-based augmentation (Images © Google).

In contrast, augmented reality applications utilize a user's physical surroundings as canvas for the content displayed directly on the smartphone [201]. While some examples such as Google's Ingress simply utilize a map to display in-game content to users, others, such as Pokémon Go, make use of the smartphone camera, as illustrated in Figure 1. In the latter case, content is directly displayed within a live video of the real-world surroundings. In contrast to virtual reality applications, a user actively explores content by moving around in the real world.

As the application displays content based on actual physical surroundings, user mobility is essential when interacting with the game. This characteristic motivated distinct mobile augmented reality applications for health [38]. Such applications aim

*Physical
mobility*

to trigger the user to perform a given activity by including certain elements of games, referred to as a *gamification* of the respective application. A study by Althoff et al. shows that the mobile augmented reality game Pokémon Go leads to a moderate increase in daily activity [3]. However, according to another study by Howe et al., this effect attenuates after a few weeks. Still, the combination of augmented reality and physical mobility leads to interesting interaction characteristics among the users of a mobile augmented reality game, as discussed in the following section.

2.1.2 Interaction and Attraction Characteristics

Users of mobile augmented reality games interact with *i*) other users (in proximity and globally), and *ii*) specific points of interest defined in the game. While the actual gameplay elements may vary depending on the game at hand, users interacting with the game access and manipulate a shared game world and its state. Consequently, actions of a single user influence the gameplay of other users. Figure 2 illustrates this interaction pattern between users and with Points of Interest (PoIs), as discussed in the following.

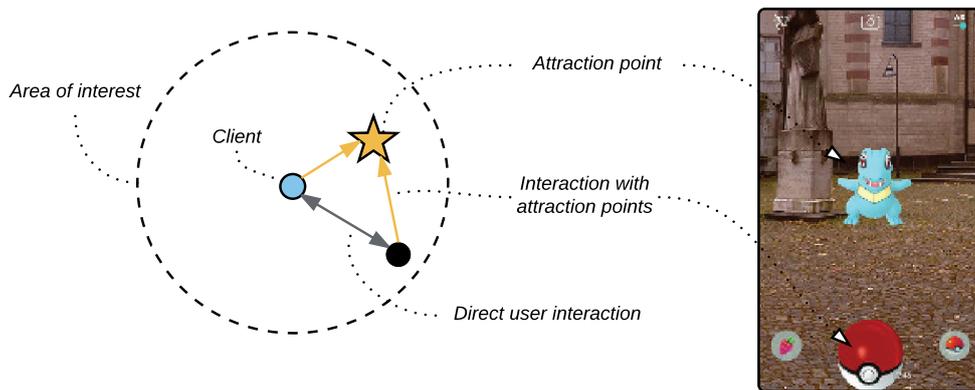


Figure 2: Basic terms and interaction patterns in mobile augmented reality games.

Interaction between users

Users can interact with each other over larger distances through in-game chats. If users are within close proximity, game-specific direct interactions between these users become possible. The set of possible interactions can range from in-game transactions of goods or valuable items to interactive fights that rely heavily on the augmentation through the smartphone’s camera. It is important to note that interaction between users does not necessarily have to involve the application itself. Instead, users might simply communicate verbally when in close proximity or by utilizing a third-party messaging application. Even if the communication itself does not take place within the application in such cases, it can still influence the game. These effects range from sudden movement of a local group of users to coordinated global actions with great impact on the game state. Examples include global operations in Google Ingress, where players around the world coordinate their actions to score points for their faction. In “Operation Green Marble”, for example, thousands of

players coordinated their actions to *win* the northern hemisphere for their faction in the augmented reality game Ingress.¹

As mentioned, the real world is used as a canvas for the contents of the mobile augmented reality game. Consequently, real-world features such as streets, sights, and public places are incorporated into the gameplay as elements users can interact with. In Twostone [38], for example, public walkways are used to determine the structure of the playing field. In Google Ingress, public sights and historical buildings serve as so-called *portals*, in-game entities that users interact with. Within Pokémon Go, the probability of encountering and catching rare Pokémon is higher at specific locations, often at public parks or recreational areas. Independent of the game and regardless of the possible interactions, these elements can be classified as PoIs. PoIs attract users due to the expected gain that is achieved when interacting with them in the game. Consequently, PoIs and their attractiveness within a game influence the mobility of users, and, in turn, on the utilization of the underlying communication system [84]. In this thesis, we use the term PoI and *attraction point* interchangeably.

Points of Interest

To model the interaction between users and the interest-based communication in mobile social applications, we introduce the publish/subscribe communication paradigm in the following section.

2.2 EVENT BROKERING AND THE PUBLISH/SUBSCRIBE PARADIGM

Figure 3 illustrates the general communication pattern of mobile social applications, modeled with the publish/subscribe paradigm. The cloud acts as a central *broker* for incoming information about users' actions, hereafter referred to as *events*. It is up to the broker to forward incoming events to all users that need to be notified of the particular event, a task commonly referred to as *event brokering*. Whether a user needs to be notified of a particular event can depend on many (usually application-specific) factors. For the case of a mobile augmented reality game, these factors include the user's physical distance to the originator of the event, for example.

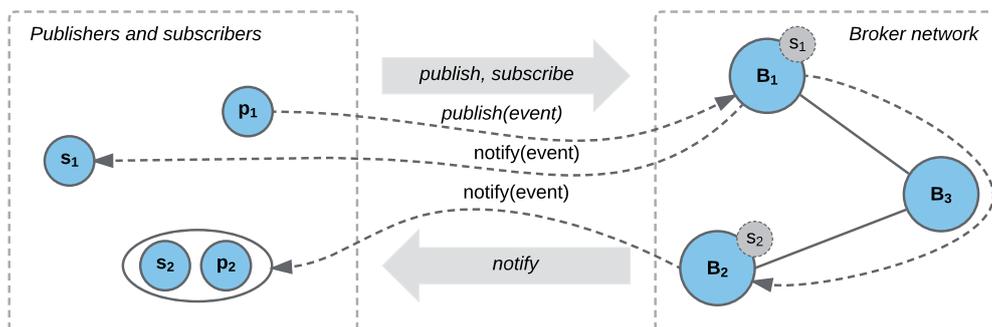


Figure 3: Illustration of the publish/subscribe paradigm.

¹ <https://heise.de/-2224133> [Accessed March 8th, 2017].

Brokers,
publishers,
and
subscribers

The publish/subscribe paradigm offers a generic and commonly used protocol design for the realization of event brokering. It defines three distinct roles: *brokers*, *publishers*, and *subscribers* [40, 180]. These roles correspond to our entities, with the cloud, as already discussed, taking over the role of the broker. Users become publishers by generating events which they send to the broker. At the same time, users can also act as subscribers, as they are interested in a specific subset of events. The publish/subscribe paradigm defines how users state this interest: they subscribe at the broker, allowing the broker to filter incoming events against these subscriptions to determine the set of users that are to be notified.

Space, time,
and synchro-
nization
decoupling

By realizing event brokering with the publish/subscribe paradigm, senders and receivers of a specific event are decoupled with respect to three properties [40]. First, they are decoupled in space, meaning that the set of subscribers is not known by the publisher of an event. Instead, the publisher simply sends the event to the broker. Second, depending on the actual implementation, publishers and subscribers do not need to be online at the same time, decoupling them in time. Instead, the broker can be configured to store events and deliver these older events once a subscriber joins the system. Third, as publishers and subscribers communicate only with the broker and not with each other, they are decoupled with respect to synchronization.

Structure of
subscriptions

The paradigm itself does not define the actual structure of subscriptions and the resulting filter operations at a broker. However, generic categories have emerged over the last years, differing in their expressiveness, intended application domain, and complexity. Existing publish/subscribe systems can be classified as *i)* channel-based, *ii)* attribute-based, or *iii)* content-based. The respective scheme defines the basic structure of events and subscriptions, and thereby the expressiveness of the subscriptions that can be issued by an application.

Channel-
based

The channel-based scheme supports subscriptions to pre-defined channels. These channels can be specified through strings or numerical identifiers, such as “stock”. Some realizations of the channel-based scheme support tree-like nesting of channels by using an URL-like structure. This would allow subscriptions to “stock” that also cover events published to “stock/google” or “stock/amazon”. Sometimes, channel-based schemes are also referred to as *topic-based*.

Attribute-
based

Attribute-based schemes enable subscriptions that filter based on a set of typed key-value pairs, referred to as attributes. Depending on the type of an attribute, different filter operations can be supported by the publish/subscribe system. For example, most attribute-based schemes support numerical filter operations like less-than, greater-than, and equal-to within a subscription. A subscription could filter, e. g., for events with a stock value greater than a given threshold: (stockvalue, 150, “>”). The syntax used for the example is explained in more detail in Section 4.2.1.

Content-based

Content-based schemes aim to provide the highest level of expressiveness by allowing filters to operate directly on the application payload. Available filters commonly include text search or support for generic regular expressions, if the application payload is rather unstructured. Other approaches assume a specific structure of the payload and thereby increase the expressiveness of filters, while at the same time increas-

ing the coupling between application and publish/subscribe system. Consequently, content-based publish/subscribe systems are often very much application-tailored.

Location-based

Recently, especially for the scenario of mobile applications, a number of location-based schemes have been proposed. This includes generic context-based schemes. They enable filtering based on additional, time-dependent attributes such as the subscribers' current locations. We discuss the respective filter schemes in detail in Section 3.1.3, given that their functionality constitutes an essential building block in a communication system for location-based mobile social applications.

However, a core characteristic of location-based subscriptions is their support for *interest mobility*, sometimes referred to as *logical mobility*. In contrast to static subscriptions resulting from the aforementioned schemes, a location-based subscription adapts itself by updating its filters to reflect the mobility of the subscriber. Consequently, the physical mobility of the subscriber is mapped to the publish/subscribe domain as a feature of the subscription scheme.

Interest mobility

In the following, we provide a brief primer on the technical aspects of communication networks and, more specifically, mobile ad hoc networks. This section provides the technical foundation for the discussion of existing publish/subscribe systems and their suitability for mobile social applications in Chapter 3.

2.3 COMMUNICATION NETWORKS

Communication networks constitute the technical foundation for the aforementioned publish/subscribe systems. A communication network connects multiple end systems via a shared communication medium [134, 179]. In this thesis, we focus on the Internet as a prominent example of a communication network, often also referred to as a *computer network*. Computer networks are standardized by the Organization for Standardization (ISO) in the Open Systems Interconnection (OSI) model [179]. According to the OSI model, a computer network is abstracted into seven functional *layers*, as illustrated in Figure 4. Each layer provides *services* to the upper layer and utilizes the respective services of the lower layer.² As illustrated in Figure 4, instances of mechanisms realized on a given layer communicate with each other via well-defined *protocols*. Therefore, they exchange so-called Protocol Data Units (PDUs), with PDUs of a higher layer being concatenated by the layer below with a protocol header, a footer, or both. The actual transmission of data happens at the physical layer. The resulting flow of data is also shown in Figure 4.

Layers and services

Protocols

Regarding the Internet, a simplified version of the OSI model is usually considered: the TCP/IP model as shown in Figure 4, named after the network protocol used in the Internet, IP, and the Transmission Control Protocol (TCP), a prominent transport protocol [179]. It is important to note that the network layer and all layers above operate end-to-end protocols between two systems. This is referred to as the *end-to-end principle*, according to which application-specific functionality is limited to the end hosts, whereas intermediate components are only utilized to establish the

End-to-end principle

² Technically, the ISO/OSI model specifies that service utilization is limited to adjacent layers. However, this rule is often (intentionally) violated to optimize for performance at the cost of loss of generality.

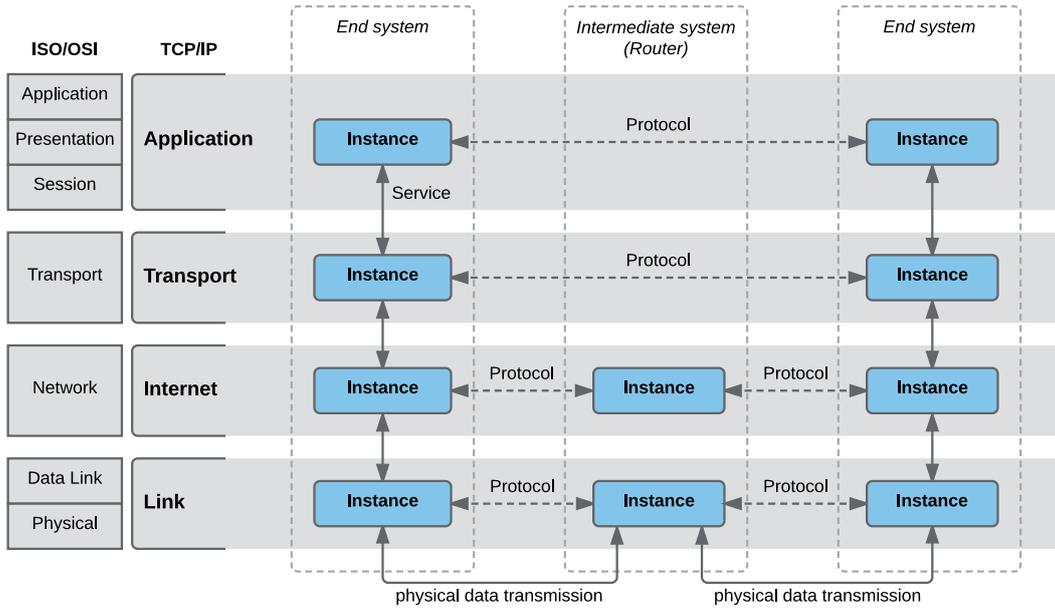


Figure 4: Layered model for computer networks.

respective connection. This principle does not hold true for the Internet, especially if mobile devices and the respective cellular networks are considered, given that a number of intermediate components, e. g., Network Address Translation (NAT) boxes, intercept the respective protocols. However, for the mechanisms discussed in this thesis, we assume end-to-end connectivity, given that mechanisms for, e. g., NAT traversal can be applied to circumvent the respective middleboxes.

In the following, we model the Internet (including the cellular network) as a *black box* that simply enables communication between mobile devices and a (cloud-based) broker network with given Quality of Service (QoS) attributes. Considered QoS attributes include the latency, bandwidth, and reliability of the respective connection.

Communication Patterns

Based on this black box approach, we now discuss the communication pattern that results from applying the publish/subscribe paradigm to location-based mobile social applications as previously discussed. Mobile clients act as producers and consumers of data (i. e., publishers and subscribers of events). A mobile client sends an event to its assigned broker, thereby communicating in a one-to-one pattern. However, as multiple clients can be subscribed to the respective event, its distribution to the set of interested clients follows a one-to-many pattern.

The one-to-many pattern can be realized as a sequence of individual one-to-one transmissions (unicasts) from the broker to each subscriber. However, this leads to redundant data transmission within the communication network, consuming significant upload capacity of the respective broker. In a distributed broker network, mul-

Unicast and
multicast

multiple brokers might be involved in notifying the respective subscribers, leading to many-to-many communication. However, as later discussed in Section 3.2.1, clients are usually assigned to brokers based on their interest (i. e., their subscriptions) to increase routing efficiency within the broker network. Therefore, we assume that a single broker acts as source for the respective notifications.

To address this issue, multicast-based data dissemination can be utilized. Here, data intended for multiple recipients is transmitted only once by the broker, and the communication network itself takes care of delivering it to all interested clients. IP multicast has been proposed as a standardized protocol that realizes this behavior by addressing a group rather than individual clients. Clients can join the respective group to receive all data that is sent to the group's address. Group management is then performed by the intermediate routers via a dedicated protocol, the Internet Group Management Protocol (IGMP) in IPv4 and Multicast Listener Discovery (MLD) in IPv6. In practice, however, IP multicast is limited to a single Autonomous System (AS) and cannot be utilized for Internet-wide multicast.³

Given that we focus on location-based applications, we briefly discuss the geocast pattern. Instead of addressing individual clients or a group of clients, geocasting addresses a specific geographical coordinate or region. In an earlier work, we studied the utilization of geocasting protocols for mobile location-based services [59]. There, we relied on direct ad hoc communication between mobile devices. We introduce the resulting concept of a MANET in the following section.

Mobile Ad Hoc Networks

The term Mobile Ad Hoc Network (MANET) is used to describe a wide range of self-organizing wireless networks with different envisioned application scenarios [27]. In general, a MANET is composed out of mobile entities that communicate with each other over a shared wireless medium in a self-organizing and decentralized fashion. Popular examples constitute Wireless Sensor Networks (WSNs) and, with increasing popularity, Vehicular Ad Hoc Networks (VANETs). According to Conti et al. [27], research interest and market relevance is especially high for vehicular applications and people-centric networks, with the latter being considered in this work from an architectural perspective. The authors further envision the combined and seamless utilization of multiple different MANET manifestations in future application scenarios, which we briefly address in Section 2.4.

Figure 5 illustrates key properties of a MANET, here formed by mobile devices (e. g., smartphones) as considered in this thesis. An individual device can communicate with other devices in range of the utilized wireless technology. For smartphone-based networks, usually Wi-Fi is utilized as underlying technology. More energy-efficient protocols, e. g., ZigBee or Bluetooth, are becoming increasingly popular for

³ Even within an AS, it is usually not available for utilization by end users. This is due to the fact that accountability and security issues are not properly addressed by the group management protocols. However, IP multicast is commonly used for services that are offered directly by the respective provider, such as IP-based television (IPTV).

applications that do not require high throughput. To ease interoperability, MANETs often operate an IP network stack, with the exception of WSNs that are running on severely resource constrained devices. Different protocols exist to support end-to-end routing over a MANET [114, 159], however, their performance severely degrades with increasing client mobility and network size [23, 107].

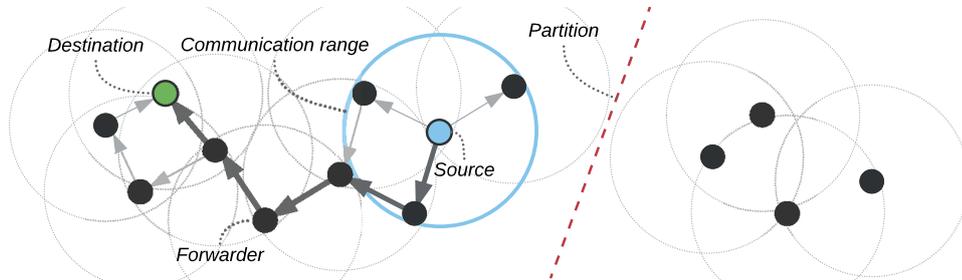


Figure 5: Multi-hop communication in a Mobile Ad Hoc Network (MANET).

To address this issue, a more decoupled approach to communication is usually used in MANETs. Multi-hop routing is addressed on the application layer relying only on single-hop unicasts or broadcasts on the lower layers. The resulting communication systems operate according to the multicast pattern, as discussed in more detail in Section 3.1.2. Depending on the application area, content is simply disseminated to all clients in the network. To this end, enabling efficient broadcasting in a MANET constitutes a research field on its own [162].

Figure 5 illustrates another issue arising in MANETs, especially if they are composed out of people carrying smartphones: partitions. Depending on the mobility and physical location of devices, areas of the network can become disconnected from other areas. These partitions might rejoin at a later point in time. This dynamic behavior of the underlying connections pose challenges to communication systems. At the same time, the behavior of participants in a MANET can also be utilized within the process of distributing content. In Delay-tolerant Networking (DTN), for example, content is intentionally stored at mobile devices and then forwarded at a later point in time, basically utilizing mobile devices as *data ferries*. While this approach utilizes client mobility to its advantage, the delay until content is finally delivered to an interested receiver increases substantially.

Hybrid
networks

Several designs exist that propose to connect MANETs with the Internet [35]. The combination is especially interesting in the context of *mobile data offloading* [138]. Here, ad hoc connectivity among mobile devices is utilized to locally exchange data (e. g., chunks of a video or prefetched web content) instead of requesting it via the cellular network. As surveyed in [138], challenges arise in the identification of potential sources and their coordination with and without central infrastructure. Additionally, potentially selfish mobile clients need to be incentivized to contribute their resources by acting as a *gateway* or relay for other clients [122, 123].

Especially in MANETs (but also in fixed networks), adaptivity to dynamic environmental and load conditions is a key requirement. This requirement is addressed with

the design concept of mechanism transitions for adaptive communication systems, as introduced in the following section.

2.4 CONCEPT OF MECHANISM TRANSITIONS

The general concept of a transition between individual mechanisms within a communication system is formalized by members of the Collaborative Research Centre “MAKI”⁴ in [46, 141, 173, 192]. In the following, we provide a summary of transition-enabled communication systems. A communication system consists of multiple mechanisms, operating on different layers of the networking stack, as previously discussed in Section 2.3. TCP, for example, is a mechanism on the transport layer. In today’s Internet, a multitude of mechanisms exist for a specific function: in addition to TCP, for example, there are several other transport protocols such as UDP or the Stream Control Transmission Protocol (SCTP). In MAKI, a group of mechanisms that provide exchangeable functionality is defined as a *multi-mechanism* [46], as illustrated in Figure 6.

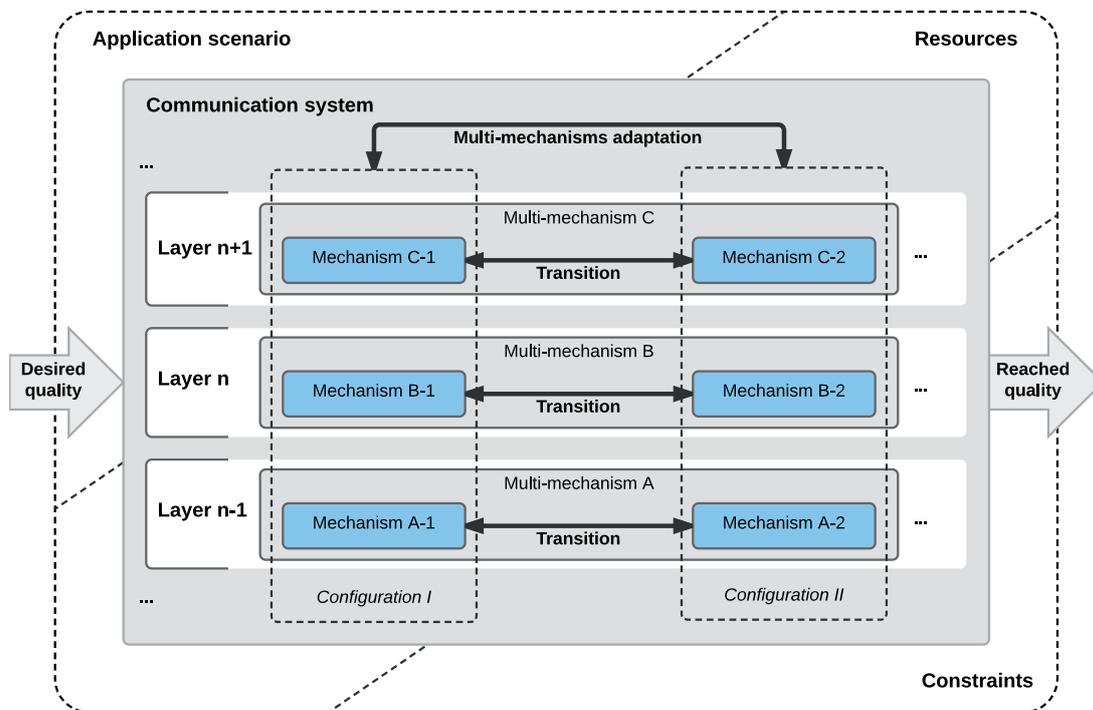


Figure 6: Mechanisms and multi-mechanisms within a communication system.

Although each mechanism within a multi-mechanism realizes the same functionality (for example, packet transport), mechanisms exhibit different performance and cost characteristics. These characteristics depend on the application scenario, the available resources, and the constraints imposed by the current environmental con-

⁴ www.maki.tu-darmstadt.de [Accessed March 8th, 2017]

ditions. Consequently, to cope with dynamic environments and application requirements, mechanisms need to be adaptive. Historically, this has led to an ever increasing number of new mechanisms.

Instead of focusing on the design of new mechanisms, MAKI aims to utilize existing mechanisms in a highly adaptive fashion. By realizing transitions between mechanisms within a multi-mechanism, the currently running mechanism can be exchanged seamlessly during operation of the communication system. By coordinating such transitions across multiple functional layers and the respective multi-mechanisms, a *multi-mechanisms adaptation* is achieved. Such a coordinated execution of transitions affecting multiple mechanisms leads to the systems switching from one configuration to another.

The set of suitable configurations is determined by the application scenario, the environmental constraints, available resources, and the desired quality. Available resources do not only pose a limitation to the transition-enabled system. Instead, they can also aid in the execution of transitions. To this end, MAKI considers the utilization of Software-defined network (SDN) for the coordination and execution of transitions affecting the available resources [65]. Thereby, the transition-enabled communication system can actively alter and influence available resources.

MAPE-cycle

Transitions can affect multiple layers and, given the distributed nature of communication systems, have different regional scopes within a system. For their coordination, a distributed control cycle is envisioned within MAKI. This process follows the steps of the well-known MAPE-cycle [85] of *monitoring* a system's or mechanism's state, *analyzing* the gathered information, *planning* suitable transitions, and finally *executing* the transitions in the network, as illustrated in Figure 7. The example shows a central coordinator that performs the analysis and planning steps of the cycle. Analysis and planning might involve a *knowledge* component containing, e. g., a model of the system that is to be coordinated or information about previously observed conditions and the respective adaptations. Monitoring involves collecting relevant information from the clients that are to be controlled as well as environmental information collected from other sources (not illustrated). Once the coordinator comes to a conclusion about the adaptation that is to be performed (or the set of transitions in our case), it notifies the respective execution component at the client. This component then takes care of the execution of transitions to adapt the system accordingly by interacting with the respective mechanisms.

As indicated in the example, the MAPE-cycle can also be realized in a fully distributed manner, with each client performing its analysis and planning steps based on local or regional knowledge.

Research in the Collaborative Research Centre "MAKI" addresses distinct aspects of adaptation mechanisms, communication mechanisms, and a design methodology for transition-enabled communication systems. Within this work, we apply the concept of mechanism transitions to the specific domain of publish/subscribe systems, as discussed in Chapter 4. We further contribute to the generalized design methodology for transition-enabled communication systems by proposing domain-independent abstractions of the concepts proposed for publish/subscribe mecha-

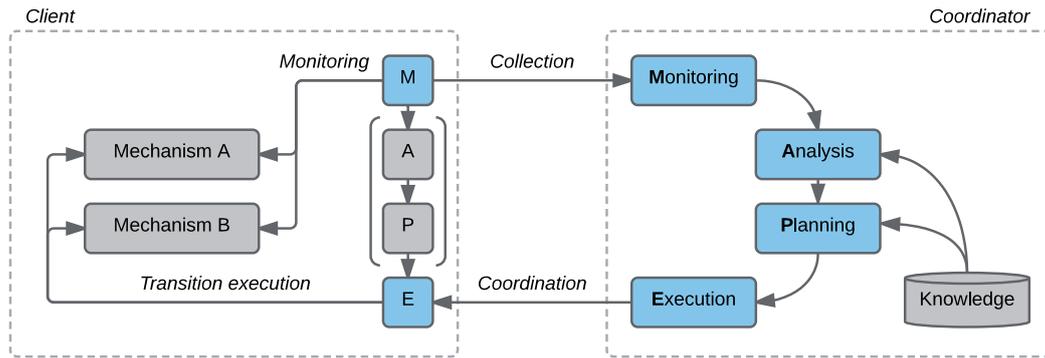


Figure 7: The MAPE-cycle for the adaptation of (distributed) systems.

nisms in Chapter 5. We evaluate the impact of coordinated mechanism transitions on the performance and behavior of a communication system, specifically focusing on the time right during and after execution. In the following chapter, we survey state of the art mechanisms and adaptation principles that motivate our design and implementation of transitions in publish/subscribe systems.

IN this chapter, we discuss the state of the art in adaptive event brokering for location-based mobile social applications. We discuss how existing mechanisms and systems address three distinct questions resulting from the scenario of mobile social applications: *i) how do publish/subscribe mechanisms address client, broker, and interest mobility?* (Section 3.1), *ii) how do they achieve adaptivity in dynamic and heterogeneous environments?* (Section 3.2), and *iii) how are transitions utilized to adapt communication systems?* (Section 3.3). The individual findings are summarized in Section 3.4, highlighting the research gap addressed in our work.

3.1 MOBILITY SUPPORT IN PUBLISH/SUBSCRIBE SYSTEMS

Mobility is an inherent and core characteristic of mobile social applications. However, in location-based applications, mobility needs to be considered with respect to two different perspectives: physical and regarding the interest. The *physical mobility* of clients and devices needs to be considered in the design of the publish/subscribe system. While mechanisms such as a client handover can ease the handling of mobile clients in publish/subscribe systems, additional means might be required to also deal with intermittent connection failures or quality degradations of the cellular connection caused by mobility. We discuss mechanisms that deal with physical mobility of clients (i. e., publishers and subscribers) in Section 3.1.1, before also considering mobile brokers in the context of MANETs in Section 3.1.2. This provides the foundation for our contributions on the locality-aware dissemination of events.

*Physical
mobility*

The client interest in content in location-based mobile social applications also depends on the physical location and, thus, is subject to mobility. The resulting *interest mobility* affects the inner workings of the publish/subscribe filtering mechanisms. While already well known in the context of interest management for Networked Virtual Environments [110], adapting the respective concepts to a mobile scenario is a challenging issue. We discuss the state of the art in supporting interest mobility in Section 3.1.3, laying the foundation for our contributions on location-based filtering.

*Interest
mobility*

3.1.1 *Producer and Consumer Mobility*

A large fraction of publish/subscribe systems was initially designed for fixed networks and, consequently, non-mobile producers and consumers of events. Examples include research prototypes such as REBECA [125], HERMES [135], SIENA [17], PADRES [105], and commercially available middleware solutions like the CORBA event

and notification services [167]. Besides, the JINI framework [4] with Apache River¹ as prominent reference implementation is to be mentioned. The main focus of these projects lies on the organization of brokers, the representation of subscriptions (often referred to as the *subscription language*), and the resulting expressiveness of the filtering process, all within a rather static environment assuming fixed networks.

However, due to the increasing number of handheld devices and new application domains, the need to support mobile producers and consumers arises. Decentralized broker networks aim at increasing the locality of event processing by bundling subscriptions [183] with similar content on a single broker, as discussed in more detail in Section 3.2. Consequently, associating clients to a broker based on their current location is an intuitive design goal, especially as in location-based services physical location and interest in content are intertwined. Given that clients are mobile, the broker network needs to support client handovers between different brokers. Such a handover process can be realized with various different mechanisms [15, 22, 24, 43, 51, 70, 126, 132], depending on the underlying system model and the application scenario, as explained in the following.

In [126], Mühl et al. extend the REBECA prototype [125] to support client mobility through *persistent connections* that handle intermittent connection failures or re-associations of mobile clients transparently to the client. To this end, the state associated to a connection (client subscriptions and the transmission queue for events) is exchanged between brokers whenever a mobile client connects to a new broker. At the same time, the routing paths within the broker network are adapted accordingly to ensure that relevant events are continued to be forwarded to the client.

Mobile clients might intentionally drop their connection to the publish/subscribe system to save bandwidth and/or battery power while the application is not actively used. Whenever the connection is re-established, the application needs to retrieve at least a subset of the previously published events for the client's current location to provide meaningful information. Cilia et al. propose an extension of REBECA's subscription model in [24] to support bootstrapping of mobile clients by replaying past events. The authors study two distinct approaches to this goal. The first approach enables clients to state a timespan when actively subscribing. The timespan determines how long events need to be stored at a broker to be replayed to a mobile client when rejoining. This requires changes to the subscription model used in REBECA and sufficient caching capabilities of brokers. The second approach introduces a virtual client acting as a cache for events on behalf of the potentially disconnected mobile client. A similar concept is proposed by Caporuscio et al. in [15] and by Fiege et al. in [43], where the authors introduce proxies acting as caches whenever a mobile client is disconnected. Events stored within a proxy are sent to the client once the connection is re-established.

The aforementioned approaches are reactive and migrate clients in the event of connectivity failures or re-associations. A proactive approach for mobile clients is presented by Ottenwälder et al. in [131, 132] in the scope of a Complex Event Processing (CEP) system for automotive scenarios. Within the CEP system, the opera-

*Persistent
connections
in Rebeca*

*Bootstrapping
mobile
applications*

*Caching
events*

*Proactive
client
handovers*

¹ <http://river.apache.org> [Accessed March 8th, 2017]

tors and their associated state information need to be migrated to brokers nearby the client's current location to maintain low delivery delays. The authors propose proactive planning of migrations based on expected client mobility and the cost of dependent migrations. Gaddah et al. propose a similar proactive scheme for mobility management in publish/subscribe systems in [51]. Here, a mobile client's subscriptions are forwarded to nearby brokers based on a neighborhood graph. At these brokers, the subscriptions remain in a passive state, meaning that they are not considered during the filtering process of the respective brokers. The handover operation activates the subscriptions at the broker that is subsequently contacted by the client as consequence of the movement. Cheung et al. further propose two algorithms to proactively migrate clients based on their current work in [22]. The handover process as such has been modeled as a transaction in [70], involving not only the edge brokers but also intermediate brokers in the broker network.

In addition to the support for client handover and migration, other mechanisms have been proposed to optimize communication with mobile consumers and producers. As a significant first step to reduce the communication overhead within an event brokering system, the brokers themselves are organized in a hierarchical and geographically distributed fashion. Thereby, processing capabilities are moved closer towards the producers and consumers. Saurez et al. propose the utilization of so-called *Foglets* within a highly distributed and localized broker platform in [163]. The authors envision a multi-tiered hierarchy, where edge and core routers possess computational capabilities to act as brokers for events produced and consumed by clients. In the proposed information model, each data object is associated with a location and filtered based on that information. The proposed system utilizes the migration model proposed by Ottenwalder et al. in [132] to deal with mobile clients.

*Bringing
brokers to the
edge*

Within the aforementioned extension of REBECA presented by Muhl et al. in [126], the authors also propose a mechanism that limits the rate of events sent to a mobile consumer. Therefore, events are categorized based on their expected importance for the client application, requiring application-specific knowledge. In case the event rate exceeds a configurable threshold, only the most relevant events are sent to the mobile client. This is a technique also applied to event brokering in wireless sensor networks [74]. The savings in terms of reduced cellular network utilization come at the cost of incomplete knowledge at the client.

*Limiting the
event rate*

In [71], Huang et al. utilize the concept of *quenching*, initially proposed for fixed broker networks by Segall et al. in [166], to reduce the number of event transmissions. The broker sends an aggregate of all currently active subscriptions to mobile producers. This enables producers to locally determine whether a new event needs to be sent to the broker network at all. However, the scheme can only operate efficiently if the set of subscriptions can be aggregated and subscriptions are static. In case of dynamic subscriptions, the aggregates would change frequently, leading to increased traffic overhead. At the same time, quenching requires mobile clients to (at least partially) execute a broker's task of filtering, leading to increased complexity

*Quenching to
reduce mobile
upload*

of the client implementation. The concept of quenching is utilized in our work on context-based subscriptions, as later discussed in Section 3.1.3.

3.1.2 *Broker Mobility in Mobile Ad Hoc Networks*

The brokers themselves can also be subject to mobility if the publish/subscribe system is deployed on a Mobile Ad Hoc Network. In MANETs, the role of the broker is distributed among the mobile devices. To reduce the communication overhead caused by wide dissemination of events, publish/subscribe systems for MANETs aim to exploit locality properties. This is true even if end-to-end connectivity is provided by a routing mechanism, as sending messages via multiple hops is a costly operation and significantly reduces the overall throughput in dense networks [107]. Consequently, most publish/subscribe systems for MANETs do not assume any underlying routing algorithm. Instead, they rely on custom mechanisms to create routing structures that cope with the dynamics and expected size of the MANET for the target application.

*Structured
overlays*

Tree-based overlay approaches for publish/subscribe have been proposed in [45, 72, 120, 121]. Additionally, many general-purpose broadcast protocols for MANETs utilize tree-like routing structures. How the respective tree structure is created, differs slightly between approaches. Ruiz et al. provide an excellent survey on the matter in [162]. When utilized for publish/subscribe systems, the tree usually connects the clients that are currently acting as brokers. Similar to the construction of the tree, the selection process for brokers differs between approaches. Friedman et al. propose to select brokers based on the neighborhood density in [45]. The authors utilize a gradient-based routing scheme to forward events and subscriptions from non-broker clients to their nearest broker. Brokers communicate with each other using a *density-biased walk* that leads to an implicit spanning tree between all brokers. While this works reasonably well in rather static scenarios with a low event rate, increased client mobility requires frequent handovers of broker functionality between clients and aggressive tree management to ensure deliveries. To this end, Mitra et al. utilize a mobility prediction model to adapt the tree structure to the predicted movement of clients in [119]. Still, the increasing overhead in mobile scenarios combined with a decreasing performance especially under higher event rates as reported in [121] renders tree-based approaches unsuitable for the scenario of location-based mobile social applications. Additionally, locality in client interaction is usually not considered, as the systems proposed in [45, 72, 120, 121] forward an event to *all* brokers. Consequently, hierarchical approaches have been proposed to utilize different routing schemes for local and global communication. Yoo et al., for example, combine regionally constrained flooding within clusters of clients with tree-based forwarding between those clusters in [199].

*Proximity-
based
forwarding*

Meier et al. propose STEAM [115, 116], a locality-aware approach to event brokering in MANETs. STEAM utilizes a proximity-based approach to event forwarding and processing, thereby combining locality-aware dissemination with location-based subscriptions discussed in more detail in Section 3.1.3. By stating an explicit range for

the validity of events, their dissemination in the network is limited. Similarly, Costa et al. propose to utilize an unstructured P2P overlay to disseminate events in a semi-probabilistic fashion [29]. Clients locally rely on a partial view of current subscriptions to decide whether to forward events or not. Other probabilistic schemes utilize Bloom filters [12] to summarize subscriptions, based on which forwarding decisions are made [198]. By using probabilistic data structures such as Bloom filters, the coordination overhead can be reduced significantly. This concept is also commonly applied in scenarios with resource constrained devices, such as sensor networks [170]. Datta et al. [32] argue that completeness is not necessarily a requirement in real-life civilian applications and propose a content-driven algorithm for data dissemination. However, the DTN-like nature of the algorithm renders it unsuitable for timely event propagation. A similar approach is also proposed by Costa et al. in [28], realizing publish/subscribe by forwarding messages based on predicted social interactions of the respective clients.

In the extreme case, each individual subscriber also acts as a broker for its own subscriptions, with events being disseminated to all clients in the network. This is also referred to as information diffusion and a plethora of protocols have been proposed, as surveyed by Ruiz et al. in [162]. We limit our discussion to representative examples that are later utilized as transition-enabled mechanisms within our work. In general, diffusion mechanisms cannot guarantee reliable delivery of events due to the possibility of network partitions in a MANET. Khelil et al. propose HYPERG [86], a hybrid gossiping-based dissemination approach that adapts to MANETs with different client densities and mobility characteristics by altering the forwarding probability accordingly. Each client periodically broadcasts a beacon that is then used by neighboring clients to estimate the current density. The authors specifically focus on handling network partitions by implementing a heuristic to retransmit messages if a partition is detected.

*Information
diffusion*

Holzer et al. add information about a client's current location to outgoing broadcasts in PLAN-B [68]. When receiving a message, clients calculate their distance to the respective sender. A hesitation time and forwarding probability is chosen based on the calculated distance, such that clients further away are preferred as forwarders. PLAN-B disseminates an event to all clients within a network. Besides, geospatial information can be utilized to forward events to a specific region rather than all clients in the network. Navas and Ko [92, 128] coined the term geocasting for the respective protocols. When utilized for location-based applications, dissemination schemes based on geospatial information can further help in lowering the load on the network [59], potentially supporting large-scale deployments as considered in [129]. An extensive survey on existing protocols is provided by Maihofer et al. in [112]. In their context-based publish/subscribe system for MANETs [44], Frey et al. rely on a geocasting algorithm to disseminate events based on the associated spatial context. Holzer et al. conduct an evaluation of different algorithms for location-based multicasting [66]. They propose a hybrid approach that disseminates queries (similar to subscriptions in our scenario) and events using a geographically scoped gossiping algorithm earlier proposed in [67]. Subsequently, matching occurs in a rendezvous-like

*Utilizing
location
information*

fashion, similar to BUBBLESTORM proposed for fixed networks by Terpstra et al. [182]. Wang et al. [188] apply the concept of geospatial routing to vehicular networks [14]. The authors explicitly focus on the direction of movement rather than the exact location. This is motivated by the fact that information disseminated in an automotive scenario is mostly relevant on a specific lane or within a specific direction of travel.

Hybrid approaches

A pure MANET-based approach is not suitable for the scenario considered in this work, as we still need to maintain global connectivity among users of a location-based mobile social application. However, the utilization of direct ad hoc communication and strategies for the distributed coordination of mobile devices are vital in achieving strong locality in event brokering. In [103], Leontiadis et al. propose a hybrid system for vehicular scenarios, where content is routed into a target area via infrastructure networks (if available). Once content reached the target area, it is continuously propagated in an ad hoc fashion for a specified amount of time. Hybrid approaches usually aim at offloading the cellular connection, as recently surveyed by Rebecchi et al. in [138]. Offloading via gateway-based infusion of content into a target area requires a selection procedure for gateways as outlined in [109, 142], for example. We utilize gateways for our hybrid dissemination mechanisms as later discussed in Chapter 4.

3.1.3 *Interest Mobility through Location-based Filtering*

Extending vs. new subscription models

In location-based applications, interest in a specific event also depends on the current location of the mobile subscriber and, potentially, on the location of the producer. Simply encoding this information within subscriptions can lead to significant overhead, as clients need to frequently alter their subscriptions to reflect their current location. According to Cugola et al. [30] and Fiege et al. [43], for example, it is inevitable to treat contextual information explicitly during event brokering. According to the authors, the flexibility of the system and its efficiency in a dynamic scenario is otherwise severely limited. In the following, we discuss how location-based filtering can be enabled *i)* by extending existing attribute- or content-based publish/subscribe systems as proposed in [13, 30, 43, 51, 78, 202], and *ii)* by introducing new subscription models for context-based filters [6, 11, 20, 41, 80, 108, 200].

Reducing filter update frequency

Fiege et al. discuss the consequences of logical mobility for the content-based publish/subscribe system REBECA in [43, 202]. The authors propose a technique to limit the overhead caused in the broker network when propagating subscription updates as a consequence of mobility. Therefore, the most accurate filter is only maintained at the broker that is responsible for the respective client. Filters propagated to other brokers in the broker network are less restrictive and forward events destined to locations that might become relevant for the client in the near future. Similar to the approach for handling physical mobility presented in [51], the authors motivate using a neighborhood graph that contains potential future logical locations to determine suitable filters.

Burcea et al. propose a dedicated location matching engine in [13] that processes events matched by the filtering engine of a traditional attribute-based publish/sub-

scribe system. If a producer or consumer is static, its geographic coordinates are directly encoded into the event or subscription as attributes. Otherwise, an identifier for the mobile client is added to outgoing events or subscriptions and the client periodically reports its current location. The location matching engine uses this identifier to retrieve the last known location of the respective client to perform location-based filtering. As location-based filtering is not part of the filter engine itself, this information cannot be utilized for optimizing filter-based routing of events in a distributed broker network.

*Additional
location
matching
engine*

Cugola et al. also separate traditional filtering from context matching in [30] by proposing an Application Programming Interface (API) to set a client's context when publishing or subscribing. The context is then distributed to other brokers via a dedicated protocol and routing entries are updated accordingly. Consequently, context information is decoupled from subscriptions, thereby reducing the overhead within the broker network. The authors state that their scheme performs well whenever the contextual filters are highly selective and rarely updated. While the former holds true within the scenario considered in this work, the latter does not apply given the dynamics of mobile clients.

*Setting
context via
dedicated
APIs*

Jayaram et al. propose parameterized subscriptions in [78]. Here, a client can use so-called *broker variables* within a subscription. Each broker variable corresponds to a time-dependent value. In case of location-based publish/subscribe, there could be two such variables: the current latitude and longitude of a mobile client. Clients issue update messages to their respective broker to update their broker variables. A similar concept is also proposed in [73]. The authors strive to maintain the scalability properties of a broker network by replacing costly re-subscriptions with more lightweight update procedures. We apply the concept of broker variables within our framework, specifically focusing on the transfer of such variables during mechanism transitions between filter schemes, as detailed in Section 4.2.

*Parametric
subscriptions*

To deal with frequent location updates, dedicated subscription models and the corresponding filter schemes have been proposed in the literature [6, 11, 20, 41, 80, 108, 200]. In all these approaches, *location-based filtering* is a key primitive of the filter scheme, resulting in increased efficiency and expressiveness in dynamic scenarios.

*Dedicated
location-based
filtering*

An intuitive representation of location-based subscriptions is proposed by Brimicombe et al. in [11] with Space-Time Envelopes (STEs). STEs describe a circular area around a client's current location, as also proposed in [41]. However, an STE is further extended with a conical shape in the direction of the client's movement. The length of the cone is adapted based on the client's movement speed. All events that are published to a location that is covered by the STE are sent to the respective client. The technical realization of STEs is discussed in more detail in Section 4.2.4. Similar concepts to represent an Area of Interest (AoI) have been proposed for interest management in NVEs [110, 127]. Relying on STEs results in distinct subscriptions for each mobile client and involves rather complex calculations to match events against the geometric shape of the envelope. To reduce complexity, other schemes propose to subscribe clients to static areas instead.

Named
locations

Chen et al. rely on named locations to match events in [20]. A named location corresponds to an area described by a polygon or circle. Events can be published to specific named locations and subscriptions filter for events based on their target location. Given that the named locations are known in advance, the resulting filter scheme closely resembles generic channel-based filtering. The authors briefly motivate that mobile clients should locally detect their current channel based on their location to reduce the load on the broker. However, they do not further elaborate on how to achieve this goal. In [80], Jodlauk et al. propose a grid-based system. Clients are assigned to a cell within the grid based on their current location. By carefully choosing the size of the respective cells, the system can be adapted with respect to the desired accuracy of event dissemination and the resulting overhead.

Minimum
Bounding
Rectangle

Other designs restrict the representation of the AoI to a rectangular shape, referred to as Minimum Bounding Rectangle (MBR) [19, 106, 200, 204]. In [204], the authors focus on data structures for the representation of nested hierarchies. Here, an MBR can contain other MBRs to allow for fine-grained location-based information retrieval. The concept is extended and applied to publish/subscribe by Zhou et al. in [106, 200], now allowing overlapping MBRs. The authors propose an R-tree [61] based index structure for the resulting location-based subscriptions and devise an efficient filtering algorithm utilizing the proposed index structure. Chen et al. further discuss the utilization of a tree-based index structure for spatial top-k publish/subscribe in [19], also relying on the MBR to describe objects and regions of interest. In their work, instead of notifying a client of all matches, only the top k results according to a given cost metric are to be forwarded. Supporting top-k queries (or stateful operators in general [33, 95]) does not lie within the scope of our work. Still, it might be a promising technique to reduce transmission of less relevant events to mobile clients on a semantic level. Given the results of the performance analysis presented in [19, 106, 200, 204], it can be assumed that existing MBR-based schemes as utilized in our work can further benefit from more elaborate index structures. Research on location queries and databases for moving objects, as surveyed in [75], can further aid in realizing efficient filtering at a broker.

Spatial top-k
queries

Language
support

Holzer et al. introduce *context aspects* [69], arbitrarily complex filter expressions that operate on locally available context variables. The authors enable location-based filtering by performing the respective distance calculations within such an expression. However, their work is focused on the practical aspects of integrating context aspects into the EVENTJAVA [42] language.

Location-based mobile social applications require support for physical mobility *and* interest mobility, while at the same time exhibiting strong locality properties in the interaction between users. However, none of the aforementioned approaches supports both, physical mobility and interest mobility, and the consequences of a combined utilization are not yet studied. Additionally, the state of the art in exploiting locality properties in mobile scenarios focuses on pure MANETs, although they can also be beneficial in a hybrid environment relying on cloud-based brokers. Supporting physical mobility and interest mobility at the same time and in a locality-aware

fashion requires an adaptive approach due to the dynamics of the scenario discussed in Section 2.1. Hence, we discuss existing approaches to adapt publish/subscribe systems to dynamic environments in the following.

3.2 ADAPTIVITY IN PUBLISH/SUBSCRIBE SYSTEMS

In the following, we provide a broader perspective on techniques used to adapt publish/subscribe systems and discuss their applicability for the scenario of location-based mobile applications. Existing approaches for adaptivity in publish/subscribe can be categorized as *broker-centric* or *client-centric*. Broker-centric approaches focus on adaptation techniques applied within broker networks, usually transparent to clients. Client-centric approaches aim at utilizing the resources of potentially mobile clients during filtering and distribution of events.

*Broker-centric
vs.
client-centric
approaches*

3.2.1 *Broker-centric: Adaptivity in Distributed Broker Networks*

Techniques such as subscription bundling [183] and interest clustering based on filter similarity [79, 102, 104, 124] have been proposed for distributed broker networks. These techniques are used to limit forwarding of events within a broker network by grouping clients with similar interest at brokers that are directly connected. Additionally, by bundling subscriptions and utilizing filter-based routing, events can be dropped early based on a broker's knowledge about the aggregated interest of each neighboring broker.

Li et al. propose a representation of attribute-based subscriptions as modified binary decision diagrams in [104], allowing a unified approach to subscription merging, covering, and routing. The authors further propose to utilize statistical information about the popularity of individual attributes and their logical relationships to adapt and optimize the routing procedure. A similar concept is also proposed by Mühl et al. in [124], significantly reducing the amount of control messages.

In their content-based publish/subscribe system BERETTA [79], Jayram et al. rely on interval-based normalization of subscriptions. During the subscription normalization, relational operators are transformed into expressions relying solely on the inclusion operator (\in). This is utilized in BERETTA's routing and matching algorithm to reduce the overhead associated with subscription updates. The authors state that their algorithm supports subscription summarization as proposed in [183].

Similar techniques have also been proposed in the field of P2P-based publish/subscribe systems [53, 133, 181]. Here, the publish/subscribe system does not provide a clear distinction between brokers and clients. Instead, a client contributes its resources to the P2P network [174]. In MAGNET [53], the core task is to construct efficient multicast groups for clients with similar interests. Therefore correlation patterns among subscriptions are utilized to group clients within an underlying Distributed Hash Table (DHT). Similarly, Terpstra et al. [181] rely on a modified version of the CHORD DHT [178] to realize content-based publish/subscribe in a P2P fashion. Besides DHTs, unstructured rendezvous-based approaches such as

BUBBLESTORM [182] have been proposed as substrate for publish/subscribe systems. Similarly, the P2P publish/subscribe system RAPPEL [133] constructs and maintains its own *friendship overlay*, grouping clients with similar interest in the underlying message substrate.

Utilizing locality in interests to achieve locality in communication is a key design decision in most P2P-based publish/subscribe systems for fixed networks. However, as shown for other application areas of P2P systems, the dynamics caused by clients joining and leaving the system poses additional challenges and limits the efficiency of the aforementioned optimization approaches for broker networks [174]. This effect is even more severe if mobile clients are expected to contribute their resources and act as brokers, as previously discussed in Section 3.1.2.

Within our work, we focus on the communication between a broker and its associated mobile clients, specifically addressing the challenges that arise within a mobile application scenario. Given our focus on location-based applications, interest in content is directly tied to a client's physical location. It is therefore reasonable to assume that the aforementioned approaches to interest-based clustering would lead to an assignment based on the location of clients. As we utilize a generic attribute-based subscription model, a broker network could still benefit from the concepts and algorithms discussed in this section. Therefore, provider-centric approaches to adaptivity can complement the contributions presented in this thesis.

3.2.2 Client-centric: Adaptivity through Reconfigurable Middleware

In order to better adapt to the dynamics of mobile scenarios, several middleware solutions have been proposed [31, 54, 66, 67, 168–170] that alter the behavior of publish/subscribe systems by actively utilizing the capabilities of client devices. These capabilities are not limited to communication interfaces for ad hoc connectivity. Additional sensor input can be utilized to derive the client's current context and adapt the middleware configuration accordingly.

Holzer et al. propose ALPS [66, 67], a location-based publish/subscribe middleware for MANETs. The message dissemination in ALPS can be parameterized with respect to the intended coverage of a message, resembling the rendezvous-based approach BUBBLESTORM for fixed networks [182]. The respective parameters of the dissemination module are adapted based on the currently observed set of events and subscriptions in the neighborhood of a client. However, the approach is not evaluated for dynamic subscriptions based on a client's current location, and the required coordination among clients produces significant message overhead. Gokhale et al. also propose to adapt the transport mechanism [54]. They follow a closed-loop approach utilizing machine learning to derive suitable rules for an adaptation of the transport mechanism based on monitoring data. In contrast to our work, the approach is currently limited to the transport protocol. We adapt multiple functional aspects of our system through the generalized design concept of mechanism transitions.

Similar dissemination mechanisms have recently also been proposed for middleware platforms for ubiquitous systems, as surveyed by Bellavista et al. in [8] and

*Adaptive
dissemination*

*Adaptation
based on
machine
learning*

Knappmeyer et al. in [89]. Based on their surveys, the authors explicitly highlight a field for future research. According to [8], systems need to “[...] self-adapt autonomously by dynamically combining most suitable data distribution methods and techniques [...] depending on current management conditions”. This motivates our research on mechanism transitions as an enabler for the envisioned dynamic combinations within a system.

With GREEN [168], Sivaharan et al. propose a middleware that is configured by composing pluggable components for different functional aspects. GREEN supports reconfigurability of components during runtime. However, the authors do not evaluate the performance impact during the reconfiguration. Instead, they focus on the memory overhead and loading time for components on the Windows CE mobile OS. The authors do not discuss a locally limited reconfiguration for a subset of clients, which constitutes an essential requirement in a heterogeneous scenario.

Component-based reconfiguration

Cugola et al. study the concept of reconfiguration during runtime with REDS [31]. The authors design REDS such that additional mechanisms can be integrated and utilized in the middleware, focusing their work on the software engineering aspects. Within REDS, functionally equivalent mechanisms are hidden behind a unified interface. Such mechanisms include local dissemination of events via ad hoc networking protocols. From an architectural point of view, the representation of functional aspects of the system as pluggable components constitutes an important step towards enabling their exchange during runtime. However, the authors neither evaluate the systems’ behavior during reconfigurations, nor the impact of client mobility.

Our concept of mechanism proxies discussed in Section 5.2.1 follows the same design goal of exposing functionality covered by multiple specific mechanisms via a unified interface. However, the *seamless* execution of transitions is a key requirement. Consequently, we propose mechanisms for state transfer and transformation, which we evaluate extensively. Additionally, we specifically address heterogeneous scenarios and enable the publish/subscribe system to adapt locally and for a confined set of clients based on application-specific mobility and workload characteristics. This is achieved by applying and extending the design concept of mechanism transitions (as introduced in Section 2.4). Therefore, we discuss approaches that apply transitions to other application domains in the following section.

3.3 ADAPTIVITY THROUGH MECHANISM TRANSITIONS

The concept of transitions between mechanisms as introduced in Section 2.4 is studied within the Collaborative Research Centre “MAKI”. The realization of transition-enabled communication systems and the applicability of generalized concepts for the execution of transitions has also been studied in other application domains. Within this section, we highlight key insights obtained from related work and our own collaborations on mechanism transitions in video streaming and network monitoring.

Video streaming systems

Based on our contributions on transitions in P2P video streaming systems [140], we explored the concept further within the live streaming systems TRANSIT [191] and ToPT [161]. In contrast to systems for scalable and adaptive Video on Demand

(VoD) streaming [1, 137, 205], a stream is only available at one source and cannot be prefetched by clients. Consequently, timely dissemination of the most recent video segments is key to achieve high Quality of Experience (QoE) [191]. In contrast to the work conducted in this thesis, we modeled transitions as local adaptation of utilized mechanisms. Instead of coordinating transitions for the whole system or a subset of clients, each client individually decides which mechanisms to use based on local knowledge. To maintain compatibility among clients without requiring global coordination, available mechanisms for scheduling and neighborhood maintenance need to operate on common messages and data types. While this limits the set of mechanisms that could be integrated into the system, it prevents additional coordination overhead for the execution of transitions [191]. Subsequent contributions highlighted the achieved adaptivity for large-scale live video streaming scenarios [160]. These approaches utilized layer-encoded video [206] to adapt the bandwidth requirements of the streaming process to the current environmental conditions. The benefits of adapting scheduling mechanisms based on user perceived quality has been modeled and evaluated by Wang et al. in [187], further highlighting the issue of cross-layer dependencies between mechanisms such as TCP and the video streaming protocol DASH.

*Parallel
utilization of
local
mechanisms*

Within TRANSIT and its extension ToPT, individual mechanisms are already separated via interfaces. However, mechanisms are not actually switched. Instead, they operate in parallel at different levels of utilization. Scheduling in TRANSIT, for example, is realized by two basic mechanisms: a pull-based request mechanism and a push-based flow mechanism. Depending on its local buffer state and the availability of data within its neighborhood, a client utilizes both mechanisms at varying degrees. In contrast, within this thesis, we execute transitions by switching between two completely separated mechanisms, without any parallel utilization. Thereby, we provide a clear distinction between a transition and a self-adaptive mechanism.

*Mobile user-
generated
video*

Other approaches explore the utilization of transitions for streaming and uploading of user-generated content [87, 151, 193, 195]. In [193, 195], the encoding scheme of the uploaded stream is altered based on the expected perceived quality on the receiving end, freeing network resources as a side effect. In [151], nearby mobile clients' resources are utilized for collaborative uploading of video streams in heterogeneous mobile environments.

*Network
monitoring*

In addition to video streaming, the concept of mechanism transitions has been proposed and explored within the domain of mobile network monitoring [153, 157, 175]. The monitoring system CRATER [157] can switch between a centralized and a distributed collection mechanism. This is motivated by the need to lower the load on the cellular infrastructure caused by background services such as monitoring, especially in situations with a high client density. Additionally, Stingl et al. motivate transitions between distinct monitoring mechanisms in MANETs in [175]. In [157], the density of clients determines whether the system switches to a gradient-based routing mechanism utilizing local ad hoc connectivity, rather than reporting results to the central entity via the mobile network.

In [153], the concept of distributed data collection is taken one step further by the utilization of gateway selection algorithms. Selecting gateways in an intelligent way

enables the monitoring system to offload traffic from the cellular network without losing completeness or reducing the accuracy. Additionally, specific fairness characteristics can be enforced to raise acceptance among clients that act as gateways. The framework of gateway selection algorithms developed in [153] is utilized in our work, as later discussed in Section 4.3.4.

*Gateway
selection for
offloading*

In addition to the aforementioned examples, transitions have been studied in the context of topology adaptation for wireless sensor networks. In [171], the authors propose transitions that alter the structure of a connectivity graph by adding and removing edges representing potential connections between sensor nodes. By removing long edges in a process referred to as *topology control* while still maintaining connectivity, the energy used to transmit data can be reduced. The concept of transitions to add and remove edges in a connectivity graph enables the generic construction of incremental topology control algorithms, as proposed in [88].

3.4 SUMMARY AND IDENTIFIED RESEARCH GAP

We previously discussed related work with respect to three questions motivated by our scenario. In the following, we briefly summarize key findings for each question and highlight the research gap addressed in our work.

How to support client, broker, and interest mobility in publish/subscribe?

Individually, each aspect of mobility has been studied extensively, as discussed in Section 3.1. Approaches dealing with client mobility focus on handling intermittent connection loss, enabling client handovers between brokers, and ensuring delivery of events during handover. Related work on broker mobility is limited to MANETs, where a subset of the mobile clients acts as brokers. The goal of disseminating events to all interested clients is tackled by structured and unstructured approaches, each with its respective advantages as discussed. In addition to physical mobility, support for interest mobility is a key requirement for location-based applications. This can be achieved through subscription models and filter schemes that support location-based filtering. In our work, we study the combination of mechanisms for location-based filtering and mechanisms for locality-aware event dissemination that tackle broker and client mobility. Furthermore, we propose mechanism combinations that offload the cellular connection at virtually no additional cost by benefiting from the filtering process of a publish/subscribe system.

How to achieve adaptivity in dynamic and heterogeneous environments?

As discussed in Section 3.2, most contributions addressing adaptive publish/subscribe systems focus on filtering and routing within the broker network. To this end, filter-based routing, subscription summarization, and filter merging allow the system to adapt to dynamic workloads. However, these techniques only benefit the broker network and do not address mobile clients. To this end, a reconfigurable middleware for publish/subscribe in MANETs aims to incorporate a mobile client's context and capabilities. Existing approaches reconfigure the middleware during runtime to

adapt to dynamic environments. However, the impact of such reconfigurations on the system's performance was not studied so far. Furthermore, reconfigurations always affect all clients within the respective network. This limits the adaptivity with respect to heterogeneous client dynamics arising in mobile social applications. In our work, we propose adaptation based on application-specific attraction points for a regionally confined group of clients, focusing explicitly on the *last mile* between broker and mobile client. We study the adaptivity of publish/subscribe systems during runtime, focusing on the impact of our proposed mechanisms on the performance and cost characteristics of the overall system.

How are mechanism transitions utilized to adapt communication systems?

Adapting systems by executing transitions between individual mechanisms proved promising for other application domains, e. g., video streaming and network monitoring. Mechanism transitions enable the utilization of state of the art mechanisms previously reviewed in this chapter. Therefore, applying the design concept of mechanism transitions to publish/subscribe is a promising approach towards achieving adaptivity under dynamic conditions. Additionally, it allows us to utilize and combine mechanisms for location-based filtering and mechanisms for locality-aware event dissemination, going beyond the capabilities of the previously discussed middleware solutions. We extend the concept of transitions to execute mechanism transitions for a subset of a broker's clients, addressing the heterogeneity of conditions in mobile social applications. We further propose and evaluate mechanisms for seamless state transfer that maintain a constantly high performance during transitions.

In this chapter, we identified and surveyed mechanisms that enable *i)* location-based filtering or locality-aware dissemination and *ii)* adaptivity to dynamic environmental conditions. However, the state of the art does not *combine* mechanisms for location-based filtering and locality-aware dissemination, which is an essential step for efficient communication in location-based mobile social applications. Furthermore, adaptivity is either limited to broker networks or cannot deal with heterogeneous application-specific mobility characteristics. In our work, we design seamless transitions between mechanisms for location-based and locality-aware publish/subscribe, addressing the requirement of fine-grained adaptivity to heterogeneous conditions arising from location-based mobile social applications. As motivated, we address the design of transitions in a mechanism-specific manner in `BYPASS.KOM`, presented in Chapter 4, and generalize our findings in the `SIMONSTRATOR.KOM` platform discussed in Chapter 5.

IN our analysis of related work, we identified a gap in the combined utilization of mechanisms for location-based filtering and locality-aware dissemination of events. As discussed in Chapter 3, applying these mechanisms to the dynamic scenario of location-based mobile social applications is further hindered by their limited adaptability. We designed BYPASS.KOM [147] as a framework to study the potential of mechanism transitions for location-based filtering and locality-aware event brokering. BYPASS.KOM enables us to include a wide range of existing mechanisms identified in our literature survey when studying the impact of transitions. At the same time, it allows us to combine mechanisms that alter the structure of the publish/subscribe system with mechanisms that alter the content of publications and subscriptions. Enabling this combined utilization of mechanisms is essential to address both, location-based filtering and locality-aware event dissemination.

In the following, we provide a brief conceptual overview of BYPASS.KOM. In Section 4.2, we present transitions between filter schemes as a promising methodology to support adaptive location-based filtering. In Section 4.3 we focus on transitions between mechanisms for locality-aware event dissemination, including ad hoc dissemination mechanisms and their utilization in hybrid, cloud-based scenarios. We further discuss the potential of transition-enabled gateway selection algorithms within such hybrid scenarios. The coexistence of mechanisms for location-based filtering and locality-aware event dissemination is finally discussed in Section 4.4. BYPASS.KOM plays a vital role in understanding the implications of mechanism transitions in a publish/subscribe system in the dynamic scenario of location-based mobile social applications. Although the mechanisms and transitions in BYPASS.KOM are specific to the domain of publish/subscribe, their understanding enables us to derive concepts for the design and evaluation of transition-enabled systems regardless of the application domain. These generalized concepts are detailed in Chapter 5 as part of the SIMONSTRATOR.KOM platform, which forms the basis for the evaluation of BYPASS.KOM presented in Chapter 6.

4.1 CONCEPTUAL OVERVIEW

BYPASS.KOM provides generic abstractions for two distinct types of mechanisms: *filter schemes* for location-based publish/subscribe and locality-aware *dissemination mechanisms*. This is illustrated in Figure 8, with highlighted components being discussed in this chapter. A client application interacts with BYPASS.KOM through an API for location-based publish/subscribe, allowing the application to subscribe to events based on the client’s current location. This API and the resulting structure of sub-

scriptions issued by client applications is further discussed in Section 4.2. The API call is then processed by the client component of a filter scheme, as highlighted in the figure. This step might involve registering for location updates with the client’s mobile OS. The subscription is then forwarded to the broker via the cellular network. Context associated with the subscription, e. g., the current location of a client, is updated according to the respective filter scheme via a custom context update protocol.

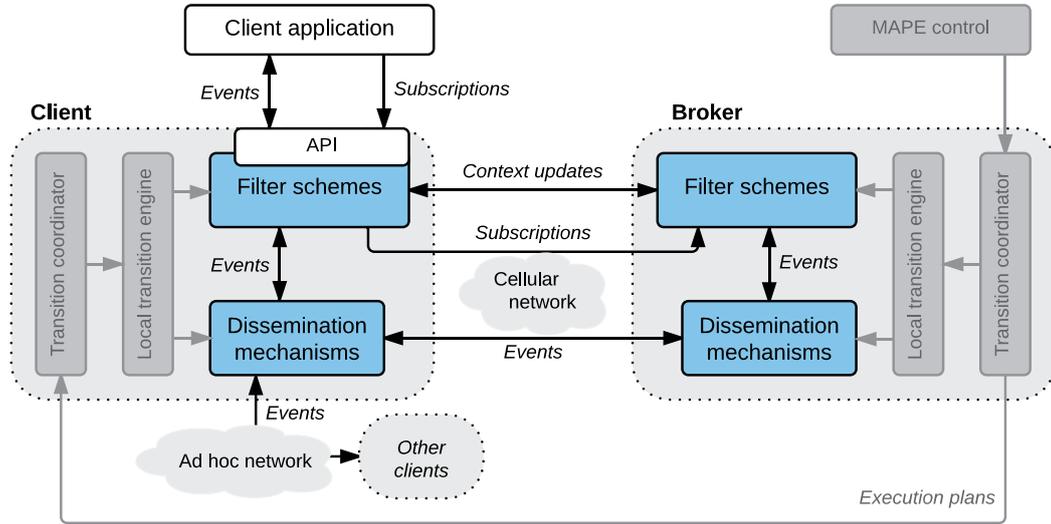


Figure 8: Overview of BYPASS.KOM and its components.

The broker uses stored subscriptions and the associated context information to filter incoming events. If an event matches a client’s subscription, the event is sent to the client via a dissemination mechanism. The dissemination mechanisms are later discussed in detail in Section 4.3. A client receiving an event notifies the subscribed client application. Depending on the utilized dissemination mechanisms, it might further forward the event to other nearby clients via an ad hoc network. Similarly, events produced by a client are handled by a dissemination mechanism. Depending on the mechanism, the event is distributed via an ad hoc dissemination mechanism or sent to the cloud-based broker.

Filter schemes and event dissemination mechanisms are designed as *transition-enabled* mechanisms, allowing seamless transitions from one mechanism to another. In this chapter, we propose a methodology for the design of transition-enabled mechanisms and apply it to filter schemes and event dissemination mechanisms. Figure 8 includes additional components for the coordination and execution of transitions, such as the *transition coordinator* and a *local transition engine*, shown in gray. These components constitute a generalization of the concepts discussed in this chapter, realized in the SIMONSTRATOR.KOM platform. They are discussed in detail in Chapter 5.

Transition types

Within BYPASS.KOM, we define three distinct types of transitions. A *total transition* is used to completely switch from one mechanism to another, for example to switch to a different filter scheme. This is modeled and discussed in detail in Section 4.2.3. However, if the filter scheme is only to be switched for a subset of subscribers, the

broker needs to maintain parallel operation of multiple filter schemes to still ensure proper operation. The respective transition is termed a *partial transition*, described in detail in Section 4.2.3. This is an essential functionality to adapt to local dynamics and spatially distributed heterogeneities arising in the scenario of location-based mobile social applications. Lastly, we propose a unified approach to parameter reconfiguration or state modification of a mechanism by modeling the respective action as *self-transitions*. A self-transition does not switch between mechanisms, but instead alters state (including configuration parameters) of the current mechanism, as later discussed in Section 4.2.5.

4.2 TRANSITIONS BETWEEN LOCATION-BASED FILTER SCHEMES

A filter scheme describes how events are matched against subscriptions. It therefore defines a model containing the structure of subscriptions and the available filter operations. For traditional publish/subscribe systems with static subscriptions, the resulting tasks for clients and brokers are clearly separated. Clients publish events and subscribe to information according to the expressiveness provided by the subscription model. Subscriptions and events are forwarded to the broker, which then determines the set of interested subscribers. These subscribers are then notified of the respective event.

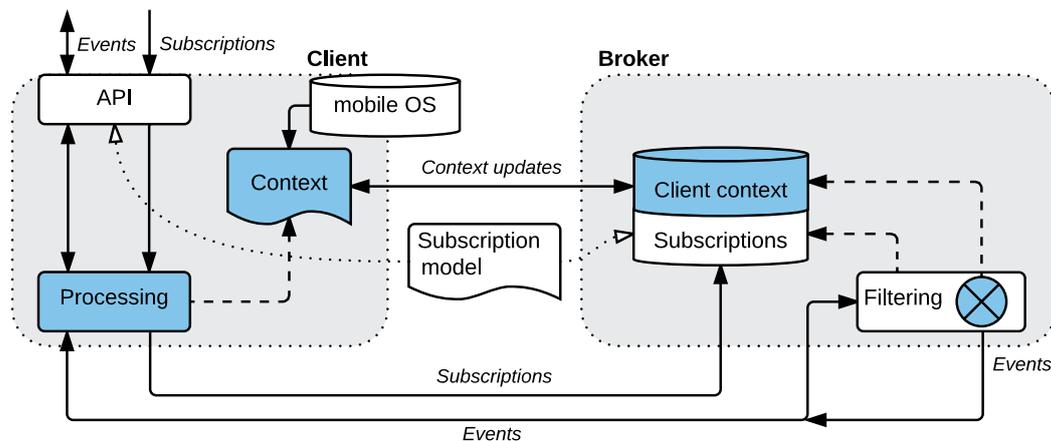


Figure 9: Location-based filter schemes relying on a custom context update protocol.

For location-based filter schemes the clear separation between client and broker no longer holds, as up-to-date information about the locations of clients are required to determine the subscribers to an incoming event. Instead, an additional protocol between client and broker is required to update state information associated with the subscription. This state information, e. g., the current locations of subscribers, needs to be maintained by the broker alongside the set of subscriptions. Consequently, a filter scheme for location-based publish subscribe additionally consists of the components highlighted in Figure 9 to provide means to update and maintain client context

State and
context
updates

at the broker and utilize that context information to filter events. Note, that these additional components are not required if the location information is already part of the subscription model. This is the case for some early approaches to location-based publish/subscribe that extend the content-based subscription model. However, as discussed in Section 3.1.3, a decoupling of the subscription model and associated contextual information is beneficial and desired, especially in dynamic scenarios.

Regarding our goal of seamless transitions between filter schemes, a common subscription model for all filter schemes is furthermore required to be able to decouple the application logic from the filter scheme that is currently utilized. In the following, we present the API and underlying common subscription model of BYPASS.KOM as a foundation required to encapsulate filter schemes and, finally, execute transitions between different schemes.

4.2.1 Model and API for Location-based Subscriptions

Regardless of the filter scheme, a common Application Programming Interface (API) for location-based publish/subscribe needs to be defined and provided to the application. To maintain compatibility with traditional attribute-based publish/subscribe systems, we extend the L1-API proposed by Pietzuch et al. [136] with specific methods for location-based publish/subscribe as proposed by Eugster et al. [41]. Instead of encoding information required for location-based publish/subscribe (i. e., the current location of a user and the size of the area of interest) within the subscription, we provide explicit methods for location-based subscriptions and events. Thereby, the context (location) is decoupled from the static content of the subscription (i. e., additional attributes) instead of being part of the subscription model.¹ According to Cugola et al. [30], this leads to increased efficiency and extensibility of the underlying communication model. The resulting API for location-based publish/subscribe and the subscription model used in BYPASS.KOM are presented in the following.²

Subscription Model

The subscription model includes *events*, *operators*, and *filters* as defined in the following. Due to the decoupled handling of context information for location-based subscriptions, the subscription model proposed in this thesis does *not* include any additions for location-based publish/subscribe.

An event e consists of attribute-value tuples $\{(a_1, \text{value}_1), \dots, (a_n, \text{value}_n)\}$ and additional application payload. We use $e_i.a$ to refer to the attribute a_i , and $e_i.value$

¹ The `subscribe` method proposed by Pietzuch et al. relies on `filter_expr`, a “[...] filter expression in any filtering language [...]” [136]. The subscription model is, thus, not defined as part of their L1-API. However, in their L3-API, the authors define an XML-based data model that relies on XPath queries for filtering. Using this model for location-based publish/subscribe would enforce tight coupling between context and static content of a subscription.

² Some shorthand and utility methods are omitted for brevity. A complete documentation of the API available in the SIMONSTRATOR.KOM platform is provided online: www.simonstrator.com [Accessed March 8th, 2017].

to refer to the associated value $value_i$ within the i -th tuple. Additionally, $e.a$ is used to refer to the set of all attributes $\{a_1, \dots, a_n\}$ contained in the event. An application creates attributes through the API. Thereby, the representation of an attribute used within the publish/subscribe system is decoupled from the logic of the client application. Consequently, the full set of available attributes A is defined by the application.

Events and attributes

An individual attribute $a \in A$ consists of a tuple (name, type) of the name and the data type of the attribute. To ease readability and prevent potential ambiguity for the application programmer, we require names of attributes to be unique regardless of the associated type. Therefore, the type of the attribute is sometimes omitted for readability, and the attribute is simply referred to by its name. As introduced for events, the name and type of an attribute is referred to using $a.name$ and $a.type$, respectively. Attributes can be instantiated with values of the respective type as a tuple: $(a, value)$. As a shorthand, the notation $(a.name, value)$ is used hereafter. An integer attribute “level” with a value of 42 is, thus, referred to as $((\text{“level”}, \mathbb{I}), 42)$, or, using the shorthand notation: $(\text{“level”}, 42)$. BYPASS.KOM supports numerical (integer and floating point) and string-based attributes and the corresponding operators, as discussed in the following. Additional types can be added by the application, given that the corresponding operators are also provided.

An operator op identifies the function (or expression) that is to be executed by the broker when filtering events. The implementation and availability of a given operation depends on the type of the attribute. In contrast to attributes, operators with the same operation name can exist for different types. As a shorthand, we therefore use op_{type} to refer to the realization of the operator op for the attribute type $type$. The operator “ $\geq_{\mathbb{I}}$ ”, for example, identifies the *greater than or equal* function for integer numbers in the prototype implementation of BYPASS.KOM. The set of available operations for a specific attribute type is referred to as OP_{type} .

Operators and filters

Combining operators and attributes leads to the specification of filters. A filter f is defined as the tuple $(a, value, op)$ of attribute, value, and operator. As in the previous cases, entries within the tuple are addressed as $f.a$, $f.value$, and $f.op$. To probe an input value against the respective filter, the function referred to by $f.op$ needs to be executed at the broker. This is denoted through the function $eval(f.op, f.value, value)$, which evaluates the operation for the provided candidate $value$ and returns a boolean indicating a match. Note, that the type of $value$ must match $f.a.type$, and that $op_{f.a.type} \in OP_{f.a.type}$ must hold. Following our example, a filter for all events where the attribute “level” is greater than or equal to 12 is defined as $(\text{“level”}, 12, \text{“}\geq\text{”})$.

Consequently, a subscription s is defined as a non-empty set of filters $s = \{f_1, \dots, f_n\}$. To match an event e against a set of subscriptions S , Algorithm 1 is applied for each subscription $s \in S$. The algorithm first checks if the attribute used within each filter f_i is also contained within the event, as the event is otherwise not covered by the subscription. As the $eval$ function is applied to all attribute-value pairs with the same attribute, a subscription can contain multiple filter entries for the same attribute. The algorithm returns true if, and only if, all filter entries are matched successfully.

Matching events and subscriptions

Finally, the clients associated to the set of subscriptions $S' \subset S$ for which algorithm 1 returned true are notified of the event e .

Algorithm 1 : Matching an event against a single subscription.

Data : Event e and subscription s **Result** : Boolean indicating a match

```

1 for  $f_i \in s$  do                                     // per definition:  $s \neq \emptyset$ 
2   if  $f_i.a \notin e.a$  then
3     return false;
4   for  $e_j \in e$  do
5     if  $e_j.a = f_i.a$  then
6       if not eval( $f_i.op, f_i.value, e_j.value$ ) then
7         return false;
8 return true;

```

Subscribing to Events around a Client's Location

As the subscription model in BYPASS.KOM does not contain any specific extensions for location-based subscriptions, we provide this functionality through a dedicated API method:

```
sub_handle subscribe(subscription, callback, location_request, radius)
```

In comparison to the `subscribe` method defined in [136], we added the underlined arguments `location_request` and `radius`. Subscription refers to a set of attribute-based filters, as defined in our subscription model. The callback is notified, whenever a matching event is delivered to the client. The method returns `sub_handle` as a unique identifier for the subscription that can later be used to unsubscribe.

As motivated, we require filter schemes to manage a client's location context themselves. Therefore, the application does not subscribe to a static location, but instead describes how continuous location updates should be handled by the filter scheme. Subscribing to a *static* location does not require any context management and can simply be achieved by adding a custom attribute type and the corresponding operator to BYPASS.KOM. Thus, it can be realized independent of the utilized filter scheme through the default subscribe method as defined in [136]

To enable an application to control the desired accuracy and frequency of updates, we demand it to pass a `location_request` containing this information. The location request is motivated by the design of the location sensor API of Google's Android mobile OS, which includes a corresponding `LocationRequest` object.³ The `location_request` simply contains the desired frequency of location updates and the accuracy of the location information required by the application, usually specified as either *low*, *medium*, or *high accuracy*. For high accuracy or high frequencies, GPS or Wi-Fi fingerprinting is utilized by the Android OS. In the case of mobile augmented reality games, high location accuracy is a key requirement.

³ The full API of Android's `LocationRequest` is documented at <https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest> [Accessed March 8th, 2017]

We assume that applications are interested in a circular AoI around the client's current location. The size of the AoI is defined by `radius`. Although being a simple integer value, this information is not encoded as an attribute within the subscription. As previously motivated, we separate contextual information associated with location-based publish/subscribe from our subscription model. Adding this client-specific attribute would hinder efficient merging of otherwise equal subscriptions within the (potentially channel- or attribute-based) broker network. Furthermore, many filter schemes utilize a custom representation of the AoI to deal with location uncertainties and low update frequencies [11]. This filter scheme-specific representation might deviate from the circular AoI used by the application through our API. It is up to the respective filter scheme how the radius as defined by the application is stored, communicated, and utilized within the filtering process, as later discussed in Section 4.2.2. We discuss the integration of such filter schemes into `BYPASS.KOM` in Section 4.2.4, demonstrating the applicability of our design.

Encoding the radius

Revoking a Location-based Subscription

The signature of the `unsubscribe` method is left unchanged compared to [136]:

```
void unsubscribe(sub_handle)
```

However, if `sub_handle` refers to a location-based subscription, the client additionally checks whether it has to stop the associated request for location updates. This procedure depends on the client-side implementation of the utilized filter scheme and is therefore discussed in Section 4.2.2.

Publishing to a Location

In addition to supporting *traditional* events, `BYPASS.KOM` supports publishing an event that is relevant at the client's current location or at an arbitrary location. Publishing an event at the client's current location is performed by a dedicated API method:

```
void publish_local(event)
```

To publish an event to an arbitrary location, `BYPASS.KOM` further defines an extension of the `publish` method that accepts a geographical location.

```
void publish(event, location)
```

The distinction between publishing events to the client's current location and publishing to arbitrary locations is essential for efficient filtering, especially if channel-based filter schemes are utilized. This is further elaborated in Section 4.2.2. In both cases, the location is *not* part of the attributes defined in the event. Instead, its handling is solely defined by the respective filter scheme. Only events published through one of the aforementioned methods are subject to location-based filtering at the broker. All other events are filtered based on their static attributes.

Own vs. arbitrary locations

4.2.2 Encapsulation of Filter Schemes

The filter schemes themselves are encapsulated as mechanisms at clients and brokers, as discussed in [151]. According to our design goals, this encapsulation has to be independent of the respective filter scheme to enable transitions. As illustrated in Figure 9, each filter scheme consists of four building blocks: *i*) a client component, *ii*) a broker component, *iii*) a protocol in between, and *iv*) a storage for client context at the broker. We explicitly provide a distinction between the broker component of a filter scheme and its storage utilized for client context. As shown by Werner et al. [190] for routing protocols, a clear separation of protocol logic and the storage of associated state and context information aids in supporting state transfer and transformation during a transition. The influence of state transfer on the performance of BYPASS.KOM is evaluated in [151] and further discussed in Section 6.2.2.

In the following, we discuss the respective building blocks for two distinct classes of location-based filter schemes identified in our analysis of the state of the art. The schemes realized and evaluated within the prototype of BYPASS.KOM are described in detail in Section 4.2.4.

Integration of Context-based Filter Schemes

Within our model, context-based filter schemes are realized as a combination of a *traditional* subscription as defined in Section 4.2.1 and context information associated to the respective subscriber or a subscription at the broker. For the case of location-based publish/subscribe, this context information consists of the last known location and the AoI of the subscriber. Incoming location-based events are matched against the set of location-based subscriptions by following algorithm 1 for all subscriptions that are still valid candidates after additional matching rules for the associated contextual information have been applied. The order of these two steps depends on the implementation of the filter scheme, as the resulting efficiency is affected by the underlying data structures and the characteristics of the subscriptions [75, 200].

In addition to simply forwarding events and subscriptions to the broker, the client component is responsible for retrieving continuous updates of the mobile device's current location when issuing a location-based subscription. This is done through interaction with the mobile OS, as introduced in Section 2.1. The application specifies an expected accuracy and update frequency of location information as part of the location request provided through the API. As described previously, multiple parallel location requests are usually grouped by the mobile OS. Nevertheless, the client component has to decide when and how often location updates have to be sent to the broker to maintain correct operation of the filter scheme.

Within BYPASS.KOM a distance-based, a frequency-based, and a hybrid approach for context updates are realized. With the distance-based approach, a new location is sent to the broker as soon as the distance between the last reported location and this new location exceeds a predefined threshold. With the frequency-based approach, locations are reported periodically, regardless of the actual physical distance between

Client
component

Context
updates

two reported locations. The hybrid approach simply reports a location as soon as either one of the previous two approaches would send an update.⁴

In addition to location updates, context-based schemes might require the transmission of additional contextual data. This includes the size and shape of the AoI. For a circular AoI, clients need to report the radius for a new location-based subscription to the broker. As this information is static for a subscription, it only needs to be reported once. For advanced schemes such as STEs [11], each location update also involves transmission of a vector that denotes the speed and direction of the client's movement. While this information could also be extrapolated at the broker, direct access to sensors like compass and gyroscope lead to higher accuracy of client-reported information. Context-based subscription schemes rely on a unidirectional context update protocol. All updates are initiated by the client based on local knowledge obtained from sensor measurements.

Consequently, the broker component does not initiate any communication other than the usual forwarding of matched events to subscribed clients. Matching, as briefly discussed, involves scheme-specific operations on the stored contextual information. For the case of a circular AoI, matching simply involves calculating the distance between subscriber and target event location and comparing the result to the provided radius of the AoI. A geographical location l_i is specified as a pair $l_i = (\varphi_i, \lambda_i)$ of latitude and longitude, respectively. The distance $d(l_1, l_2)$ between two locations l_1 and l_2 is calculated at the broker using the haversine formula [184], with R denoting the Earth's radius and φ_x, λ_x given in radians:

*Broker
component*

*Distance
calculation*

$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right).$$

Within location-based applications, the radius of the AoI usually lies in the range of a few hundred meters. As discussed previously, we further assume that clients are already assigned to coarsely distributed brokers based on their location. Therefore, inaccuracies arising from calculations of very large distances using the haversine formula are not an issue within our work. As soon as schemes construct more complex shapes for their AoI, calculations are usually executed in the Cartesian coordinate space. A location (φ_i, λ_i) on the Earth's surface can be mapped to the corresponding (x_i, y_i) coordinate using the Mercator projection [113], with

$$x_i = R\lambda_i, \quad y_i = R \ln \left[\tan \left(\frac{\pi}{4} + \frac{\varphi_i}{2} \right) \right].$$

Consequently, the distance d between two locations l_1 and l_2 is now simply calculated as the euclidean distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

The Cartesian coordinate space is, for example, utilized within our prototype when calculating the conical extensions of the AoI for the STE scheme [11].

⁴ If a location is reported as consequence of the distance-based method, the frequency-based approach simply begins a new period to avoid duplicate location updates.

State
maintenance

To perform these calculations, all context-based filter schemes need to store the last known location of subscribers. Additionally, the AoI needs to be stored for each location-based subscription. If client mobility is to be extrapolated by the broker, further information such as timestamps of the last update might need to be maintained. As this additional information is decoupled from the subscription itself due to the design of our subscription model, we do not pose any restrictions on the context information associated to a subscription and/or a subscriber. Following this blueprint, we integrate state of the art approaches for context-based filtering into our prototype of BYPASS.KOM, as later presented in Section 4.2.4.

Mapping Locations to Channel-based Filter Schemes

Channel
initialization

Utilizing a channel-based filter scheme provides the benefit of increased efficiency at the broker and enables the utilization of lower-layer multicast technologies when distributing events to a larger set of subscribers, as already discussed in Section 3.1.3. To integrate channel-based schemes into BYPASS.KOM, we follow the same methodology as for context-based filter schemes by dividing the filter schemes into client and broker components that communicate via a scheme-specific context update protocol. All channel-based schemes for location-based publish/subscribe strive to solve two fundamental problems: *i)* an efficient initialization of channels, and *ii)* a lightweight assignment of clients to channels.

Channel
assignment

The initialization of channels is done by the broker component of the respective filter scheme. As discussed in Section 3.1.3, channels can simply correspond to a geometric overlay such as a grid or hexagonal cells with predefined (although not necessarily static) boundaries. However, more complex schemes can benefit from available map data in assigning clients to channels. Thereby, channels can be initialized based on specific real-world places of interest, e. g., for larger events. Channels do not need to be static. Instead, the broker might decide to alter the organization of channels depending on load or any other internal or external factor.

Context
updates

Given the complete set of channels C spanning the geographical area of responsibility of a given broker, the next task is to assign clients to one or multiple channels depending on their current location. Obviously, the assignment cannot be static in the dynamic scenario of mobile location-based applications. Once a client leaves the region of coverage of a channel, the assignment process has to be repeated for that client. We introduce the *assignment function* `assign`: $(C, l, r) \rightarrow (C', c_p)$ with $c_p \in C'$ and $C' \subset C$. The function returns the set of channels C' out of all channels C that a client needs to be subscribed to when it is interested in events that are published to location l with a radius of r . Additionally, a channel c_p that is to be used for all outgoing local events is returned. This function is realized within the broker component, as it requires full knowledge of all available channels. Depending on the filter scheme, additional (potentially application-specific) input other than the client's current location can be utilized to perform the assignment.

Knowledge about the complete set of channels C and their geographical boundaries is usually not available at the client components. Instead, a client initially retrieves a local view consisting of (at least) the assigned channels C' from the broker.

This local view can either be requested explicitly by the client, for example by sending the current location to the broker, or it can be pushed to currently subscribed clients by the broker, for example as consequence of a re-initialization of the available channels. Filter schemes aim to minimize the communication overhead associated with acquiring the assigned channels. Therefore, clients do not report their location periodically, but instead perform local tests based on C' and any additional local knowledge available, on whether they need to contact the broker to update their assignment. This is discussed in more detail in Section 4.2.4.

As discussed, the API for location-based publish/subscribe defined in `BYPASS.KOM` distinguishes between events targeted at the client's current location and events for arbitrary locations. This distinction is essential for channel-based filter schemes. If an event is targeted at the client's current location, it is published to the channel c_p . Thereby, filtering at the broker component does not involve any additional calculations related to the location information.

However, for arbitrary target locations, the channels that are to be notified are usually not known to the client. Determining the target channels for a location would require local knowledge of the broker's assignment function, which is usually not the case. Therefore, the event together with its target location have to be sent to the broker, where the respective channels are then determined by applying the assignment function to the target location. This process can introduce significant computational overhead at the broker, thereby counteracting the benefits usually associated with channel-based schemes. This motivates the execution of transitions depending on the workload characteristics. As soon as the fraction of events published to arbitrary locations exceeds a threshold, a context-based filter scheme could be utilized instead.

*Publishing to
arbitrary
locations*

Multiplexing Filter Scheme Messages

To fully encapsulate filter schemes, we also need to consider the utilization of network resources. This is especially important during a transition, where two filter schemes can be active at a client at the same time, as discussed in the following section. Therefore, filter scheme specific messages used by the context update protocol are multiplexed. The *network proxy* component at brokers and clients adds a flag to outgoing messages, indicating the currently active filter scheme of the sender. Relying on this flag, the receiving entity's network proxy component is able to determine the corresponding instance that has to be notified of the incoming message.

The events and subscriptions themselves are not scheme specific. This is due to our separation between subscription model and associated contextual information. Consequently, they are sent via a common protocol regardless of the filter scheme. This mechanism ensures that events and subscriptions issued during a transition from one scheme to another can still be processed by the broker. Nevertheless, the network proxy adds a flag to identify the currently active filter scheme. This information can be utilized to detect failed client transitions and is required if the broker operates multiple schemes in parallel to cater for different groups of clients. The latter is discussed in the context of coexisting mechanisms in Section 4.4.

*Scheme-
independent
messages*

Piggybacking
contextual
information

As most filter schemes associate context to a subscription once the subscribe method is invoked, *BYPASS.KOM* supports piggybacking of messages. If a scheme's client component wants to associate contextual information to an outgoing subscription or event, the respective context update message is simply appended to the basic message. This reduces the overhead caused by lower-layer message headers, which is especially important if context update messages carry only a few bytes payload.

4.2.3 Executing Transitions between Filter Schemes

In addition to the encapsulation of individual schemes into broker- and client components and the corresponding context update protocol, we need to control each component's lifecycle to execute transitions. By relying on generic lifecycle methods as later described in Section 5.2.1, we can start and stop specific mechanism instances during a transition without utilizing any mechanism-specific functionality. In the following, we focus on the local execution of a transition once a trigger is received, as illustrated in Figure 10. The local transition engine is a generic component that is later discussed in Chapter 5. For now, we simply assume that a transition is to be executed and that clients are notified of this decision.

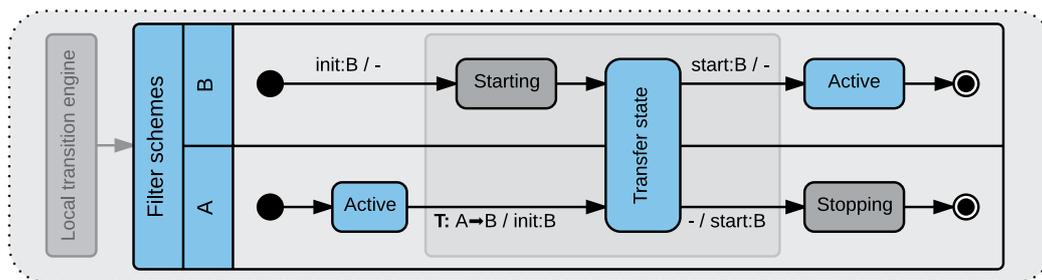


Figure 10: Execution of the filter scheme transition $T: A \rightarrow B$.

Total vs.
partial
transitions

Here, we have to distinguish between two fundamentally different scenarios resulting from the target scope of a transition. A transition can either affect all clients associated to a broker (which we refer to as a *total* transition), or just a subset of clients (a *partial* transition). In [144], we focused on total transitions, arguing that a broker is responsible for a relatively small and locally confined set of clients. The resulting consequences on the execution of a transition are discussed in the following section. Afterwards, we consider the case where transitions affect only a subset of clients assigned to a broker.

Total Transitions on all Clients

In the following, we refer to the total transition from one filter scheme A to another filter scheme B as $T: A \rightarrow B$. The process of applying T to the client component is discussed hereafter.

First, the client component of the new filter scheme B is initialized. At this stage, all interactions with the client's filter scheme are still directed to scheme A. After B is successfully initialized, relevant state information from A is transferred to B. This step might involve a transformation of the state to fit B. This process is referred to as the *state transfer* phase of T.

Relevant state of a filter scheme's client component includes the client's subscriptions and corresponding callbacks. Without transferring this information from one scheme to the other, the client application would have to resubscribe after each transition. Consequently, transitions would no longer be transparent to the application and additional overhead is caused, as shown in Section 6.2. As we rely on a common subscription model for all filter schemes, transferring subscriptions and their application callbacks does not require transformation.

*State transfer
at the client*

The state transfer phase of a transition T must not involve any network communication between client and broker components of B. By limiting state transfer to local operations, it can be realized in an atomic fashion with a guaranteed result. This is essential to maintain consistent state: if events or subscriptions are initialized by the client application during the state transfer phase, they would not be available within the new scheme B. After the state transfer took place, all application calls are directed to the new client component of filter scheme B and scheme A is discarded. At this stage, the transition finished successfully.

*Atomicity of a
transition*

Still, some schemes require additional bootstrapping that involves contacting the broker component. One example are channel-based schemes that require initial assignment of channels to the client before being able to process new events efficiently, as discussed in Section 4.2.2. This is referred to as the *bootstrap phase* of a filter scheme. As the bootstrap phase takes place after the transition has finished, it is no longer formally a part of the transition. More specifically, the bootstrap phase of a filter scheme is usually realized in the same way as an initialization of a scheme.

*Bootstrapping
after a
transition*

Executing a total transition at the broker does not differ from the previously described process for client component transitions. As the transition affects all clients, we do not need to maintain parallel operation of multiple different filter schemes. Instead, the new scheme B is initialized, state is transferred locally, and scheme A is stopped afterwards. State transfer involves moving all stored subscriptions from scheme A to scheme B.

*Transition at
the broker*

Additionally, context such as the last known location of a subscriber can be transferred during the state transfer phase. Whether such context information is available or usable during a transition depends on the respective filter schemes. Context information that is required by B but not available within A has to be gathered during the bootstrap phase of the filter scheme. This process can be initiated by the client component or the broker component, as previously discussed.

As for the client component, the state transfer phase of the transition at the broker is atomic and must not contain any network communication. However, given that some schemes might require contextual information to ensure correct location-based filtering of events, we propose three strategies to handle events that arrive during the bootstrap phase:

*Dealing with
missing
context*

FILTER WITHOUT CONTEXT. If the overall latency of the event delivery is to be minimized, we simply treat location-based subscriptions as normal subscriptions until the context information has arrived. Thereby, the event is forwarded to all matching subscribers regardless of their actual location, requiring additional filtering at the client. This leads to increased overhead due to unnecessary transmissions.

WAIT FOR CONTEXT. Here, incoming events are stored in a buffer while the bootstrap phase has not yet finished for a subscriber. Once the context information arrives, the events are filtered and matching events are delivered to the client. This method ensures that only relevant events are delivered to clients. However, the overall latency for event delivery increases.

IGNORE. This strategy ignores the respective subscription until required client context has arrived. This is only applicable if the application can tolerate potential loss of events during a transition. It does not infer any overhead or additional complexity at the broker.

In the prototype of *BYPASS.KOM*, we ignore subscriptions with incomplete contextual information. In Section 6.2.2, we show that by applying our state transfer mechanisms, we can still ensure reliable operation during most transitions. However, if reliable delivery of events is to be guaranteed, one of the other strategies is to be selected based on the tolerable delivery delay.

Partial Transitions on a Subset of Clients

If transitions between filter schemes are to be executed only for a subset of the assigned clients, this affects the execution of the transition at the broker component. Instead of applying the total transition T , the partial transition T^{\parallel} is executed at the broker as discussed in the following. Clients simply execute the total transition T , as described in the previous section.

At the broker, two or more filter schemes need to be running in parallel, if subsets of clients use different schemes as consequence of a partial transition. Incoming events need to be processed by all active schemes to guarantee delivery to all interested clients. Therefore, the network proxy introduced in Section 4.2.2 duplicates the respective incoming message and forwards it to all active filter schemes. Subscriptions issued by clients are only processed by the scheme that is currently active at the client. The flag attached to outgoing messages at the client is used by the network proxy to determine the correct filter scheme.

How a partial transition $T^{\parallel}(u) \forall u \in U'$ between scheme A and B is executed for a set of clients U' depends on whether scheme B is already running at the broker. Assuming that this is not the case, the broker component of scheme B is initialized, and the state transfer phase is started. However, in contrast to a total transition, only state that is associated to clients $u \in U'$ that are affected by the transition T^{\parallel} is transferred. As is the case for the total transition T , state includes all subscriptions of the respective clients and, optionally, contextual information managed by the filter

scheme. As soon as the state transfer is finished, scheme B is actively utilized by the network proxy as described previously and starts processing incoming events.

If scheme B is already active when a transition is to be executed, the initialization phase is skipped. In both cases, bootstrapping for the newly assigned clients can occur, as is the case for a total transition. If scheme A is no longer utilized as a consequence of $T^{\parallel} : A \rightarrow B$, its broker component is discarded. The realization of this process is described in more detail in Section 5.2.

4.2.4 Integration of Existing Mechanisms

The modular framework `BYPASS.KOM` supports the integration of existing mechanisms to research on transitions in between these mechanisms. In this section, we discuss how representative mechanisms from the state of the art in location-based filtering are integrated into our framework. In particular, we focus on the separation into client and broker component and the context update protocol. We address how the respective filter schemes utilize the common subscription model combined with custom context handling to realize their functionality in `BYPASS.KOM`. Further, we discuss alternatives for the organization of subscriptions and the utilization of contextual information in distinct filter schemes. We provide a brief discussion on how the respective alternatives affect the complexity of the filter process at the broker.

Parametric Subscriptions

As perhaps the most intuitive filter scheme for location-based publish/subscribe, we integrate parametric subscriptions [78] into `BYPASS.KOM`, hereafter referred to as `RADIAL`. Within `RADIAL`, filters $\{f_1, \dots, f_n\}$ of a location-based subscription are augmented with the parameterized filter expression

$$f_p = (d(\$(l_u), l_e), \$(radius), "\leq").$$

We refer to the function $\$(\cdot)$ as *context resolver*. When the filter is to be evaluated, the context resolver inserts the current value for the context variable. In case of $\$(l_u)$, this is the last known location of the client as stored on the broker. The location l_e is contained within the event e that is to be matched against the subscription and does not need to be resolved by the broker. The radius of the subscription is stored as associated state information for a client's subscription and is therefore also resolved using $\$(radius)$. The function $d(\cdot)$ returns the distance between two locations as defined in Section 4.2.2.

*Context
resolver $\$(\cdot)$*

To enable location-based filtering, the broker component of `RADIAL` has to maintain *i)* the last reported location of a subscriber, l_u , and *ii)* the radius associated to a subscription, radius. The last known location is valid for all location-based subscriptions of the respective client, while the radius needs to be associated to the client *and* the respective subscription.

When subscribing to a location through our API, the requested radius is reported to the broker component as part of the context update protocol. Additionally, the

Context
update
messages

client component requests the client’s current location and following updates of this information according to the location request using the API provided by the mobile OS. The respective context updates are triggered by the client as described in Section 4.2.2. Consequently, the RADIAL scheme’s context update protocol consists of a single message type, `CtxUpdateMsg`. The `CtxUpdateMsg` carries one or more key–value pairs for context variables that are to be updated at the broker. Given that the radius associated to a subscription does not change⁵ over time, it is only reported once upon creation of the new subscription.

It is important to note that parametric subscriptions as defined by Jayaram et al. in [78] and implemented in BYPASS.KOM are not limited to location-based filtering. Any contextual information can be included in a parameterized filter, given that the respective context can be resolved at the broker.

Space-Time Envelopes

As proposed by Brimicombe et al. in [11], Space-Time Envelopes (STEs) extend the concept of matching against a circular AoI by taking the client’s current movement speed and direction into account. Thereby, as discussed previously, clients need to issue less frequent location updates and the filter scheme is more robust against higher movement speeds. We refer to our realization of the scheme within BYPASS.KOM simply as STE. To calculate the area covered by an envelope in STE, the broker needs to know *i)* a subscriber’s current location l_u , *ii)* the subscriber’s current movement vector \vec{s}_u , and *iii)* the radius associated to the subscription.

Construction
of the STE

The area is then constructed as illustrated in Figure 11: the circle defined by the client’s current location l_u and the radius of the subscription is extended with a cone in the direction of \vec{s}_u . The cone’s initial width is defined by the radius, widening with an angle of 15° . Its length is given as $\alpha \cdot 60 \cdot |\vec{s}_u|$, with $|\vec{s}_u|$ being the client’s movement speed in meters per second. The factor α can be used to adapt the scheme to different movement speeds and location update intervals. Its default value is $\alpha = 1$, corresponding to a cone covering roughly one minute of unaltered client mobility.

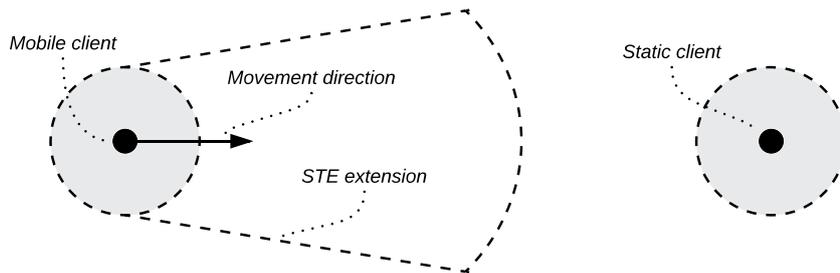


Figure 11: Area defined by a Space-Time Envelope.

To reduce the computational complexity at the broker, the respective area is calculated once after an update of any of the above-mentioned parameters and then

⁵ As defined by our API, applications specify a fixed radius when issuing a location-based subscription. To alter the respective AoI, the application simply resubscribes with an altered radius.

stored in the context storage. As the area depends on the radius associated to a subscription, the STE needs to be stored as subscription-dependent context. Thus, we refer to the STE of client u and subscription s as $STE_{u,s}$.

Incoming events are matched against location-based subscriptions by checking whether their location l_e lies within the area $STE_{u,s}$. To this end, the broker component defines a method `in_area(A, l)` that returns 1 if the location l lies within the area defined by A , or 0 otherwise. Consequently, the following additional filter is used by STE when matching events against location-based subscription:

$$f_{STE} = (\text{in_area}(\$(STE_{u,s}), l_e), 1, "=").$$

Note, that the current area of the STE for client u with subscription s is resolved as a context variable via $\$(\cdot)$.

In both schemes, RADIAL and STE, custom functions are utilized, thus increasing the complexity of the matching process at the broker. Especially for STE, determining whether a given location lies within the area covered by an STE is a rather computationally intense task. In addition to the computational complexity associated with context-based subscriptions, context variables have to be updated frequently to ensure accurate filtering at the broker. To lower the complexity of the filtering process and the frequency of context update messages, the following channel-based filter schemes are integrated into BYPASS.KOM.

*Matching
with the
in_area
function*

Grid-based Filter Schemes

Channel-based schemes aim to reduce the communication overhead caused by frequent context update messages. At the same time, they utilize pure attribute-based filtering at the broker, thereby reducing the complexity of the filtering process. Based on the general discussion of channel-based filter schemes in Section 4.2.2, we discuss three distinct representatives integrated into BYPASS.KOM. First, we discuss a basic grid with single-channel assignment, called GRID, followed by the multi-channel assignment scheme EGRID. Motivated by the characteristics of location-based mobile applications, we then propose a channel-based scheme called ATTRACT that utilizes application-specific knowledge to optimize its channel assignment.

In GRID and EGRID, channels are organized as a grid of equally sized rectangles covering the area that is managed by the broker, hereafter referred to as W . For simplicity, we assume that a client's location l_u and the bounds of W are transformed into the Cartesian coordinate space using a suitable projection as discussed before. A grid is constructed based on the MBR around the area W . This MBR is divided into equally sized rectangles, with the number of divisions for each axis being configured with the so-called *grid factor*. A grid factor of 2, for example, leads to a 2×2 grid and, consequently, 4 channels.

As defined in Section 4.2.2, the assignment function returns the respective set of channels a client needs to be subscribed to. Within GRID, a client is assigned only to the channel covering the client's current location, regardless of the AoI of any subscriptions. Within EGRID, the client is instead assigned to all channels within the MBR required to cover the AoI, as illustrated in Figure 12. Therefore, the broker

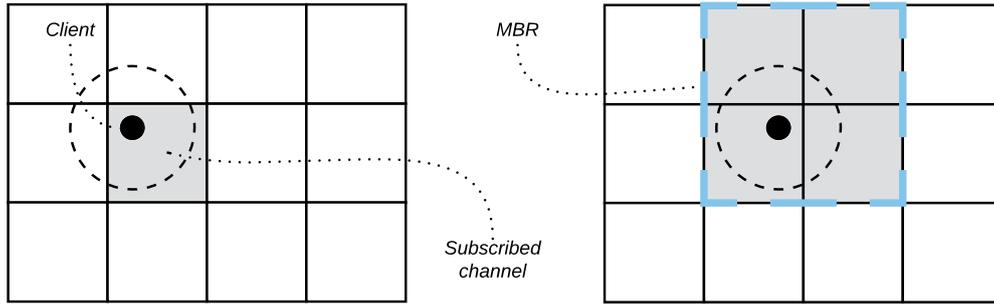


Figure 12: Channel creation and assignment in GRID and eGRID.

additionally needs to store the size of the AoI of a subscription to provide the context variable $\$(radius)$ required by eGRID. While this information is not actually required for the operation of GRID, it is nevertheless maintained as context information as it contributes to the seamlessness of transitions as evaluated in Section 6.2.2.

A client needs to send a `RequestAssignmentMsg` to the broker to be assigned to a channel. The message simply contains the client’s current location l_u , much like the aforementioned context update messages. Based on the reported location, the broker assigns the channel c_p to be used for outgoing local events and notifies the client of the respective channel name and the geographical boundaries of the MBR by sending an `AssignmentMsg`. Based on the MBR, the client can detect locally whether a new assignment needs to be requested as consequence of mobility. An `AssignmentMsg` is also issued by the broker whenever the assignment function itself is updated, for example due to an updated grid factor.

The full set of channels C' a client is subscribed to is stored as contextual information at the broker. As a client u can have multiple active location-based subscriptions with different AoIs, this context variable also depends on the respective subscription s . It can, therefore, be accessed with $\$(C'_{u,s})$. The client component adds c_p to outgoing *local* events by adding a new attribute

$$a_c = ("channel", c_p).$$

Filtering with context resolver

This attribute is used by the broker during the matching process. The broker component applies the filter

$$f_c = ("channel", \$(C'_{u,s}), "\in")$$

to all location-based subscriptions during matching. The set operation “ \in ” checks whether $c_p \in C'_{u,s}$ by comparing the respective channel names. It does *not* involve any geometric calculations. As matching does not involve a custom function in this case but instead relies on a simple hash-based comparison of channel identifiers, it is considered to be more efficient in terms of computational overhead than the previously discussed context-based schemes.

The aforementioned realization relies on the context resolver $\$(\cdot)$ to retrieve the set of channels for a client and its subscription. The same functionality can also be

realized without calls to $\$(\cdot)$ during filtering by creating $k = |C'_{u,s}|$ copies of the respective location-based subscription, with each copy s_i being assigned to one fixed channel $c_i \in C'_{u,s}$. In this case, each copy s_i is created by extending the original subscription s with a context-independent filter f_{c_i} , leading to:

$$s_i = \{f_{c_i}\} \cup s \quad \forall i \in \{1, \dots, k\} \quad \text{with} \\ f_{c_i} = (\text{"channel"}, c_i, \text{"="}) .$$

*Filtering
without
context
resolver*

With this, the context resolver is not utilized when matching incoming events against subscriptions. However, the set of subscriptions has to be altered according to the updated assignments whenever a client issues a new `RequestAssignmentMsg`, now involving the context resolver. In our application scenario, the rate of events and consequent matching operations is much higher than the rate of `RequestAssignmentMsg`, especially if channels cover a sufficiently large area. Additionally, while the total number of subscriptions increases for the context-independent subscriptions, they can be organized in data structures that enable lookup of all subscriptions for a given target channel with an average $\mathcal{O}(1)$ time complexity for non-overlapping channels [91]. In case of overlapping or nested channels, more elaborate index structures as proposed by Zhou et al. in [204] can still be applied. Compared to RADIAL and STE, this leads to location-based filtering with lower computational complexity, given that all schemes still perform attribute-based matching as outlined in Section 4.2.1.

A client cannot utilize the channel c_p as described above, if outgoing events are published to *arbitrary* locations. Instead, the event together with its target location l_e needs to be sent to the broker, which then determines the correct target channel and adds the corresponding attribute a_c . Afterwards, matching is performed as discussed above for local events.

*Arbitrary
target
locations*

The broker does not need to store any additional client context for channel-based schemes. However, to realize seamless transitions between filter schemes, the last known location of each client is maintained within BYPASS.KOM. This information is also helpful during self-transitions used to adapt the size of the grid and, consequently, the current set of channels. Self-transitions are later discussed in Section 4.2.5 and their utilization for adapting the grid sizes in channel-based schemes is evaluated in Section 6.2.3. In addition to grids, we study an application-specific approach to the assignment of channel areas in the following section.

Utilizing Application Knowledge with Attraction-based Channels

The GRID and EGRID filter schemes are application agnostic. With ATTRACT, we extend the previously discussed channel-based filter schemes by specifying an application-specific assignment function. In ATTRACT, channels are defined based on application-specific attraction points to better capture the real-world behavior of clients in mobile social applications. Attraction points are modeled as a circular area, with their center locations and radii being given, as later detailed in Section 6.1.1.

Within ATTRACT, we simply assign a channel to each attraction point rather than relying on a fixed grid as discussed in the previous section. The area covered by the

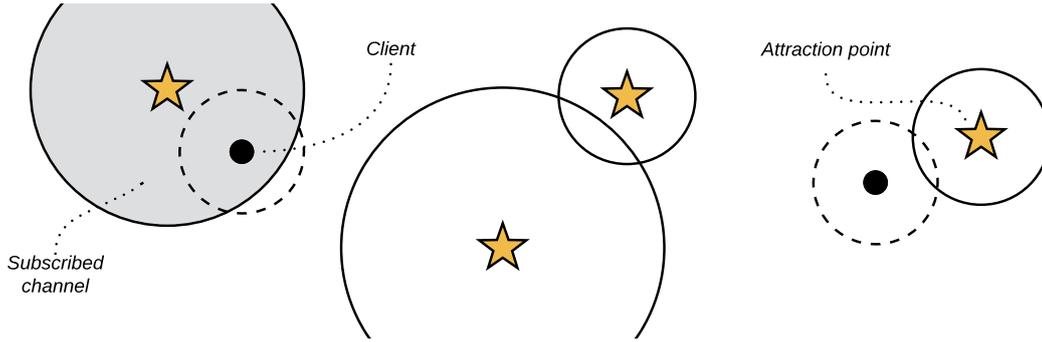


Figure 13: Channel assignment based on application-specific attraction points.

Overlapping
channels

channel is defined by the radius of the respective attraction point, as illustrated in Figure 13. In contrast to the previously discussed grid-based filter schemes, channels in ATTRACT can overlap. Additionally, depending on the locations and radii of attraction points, some locations on the map are not covered by a channel at all. In case a location is covered by multiple channels, the client subscribes to all channels. Consequently, the subscription at the broker is copied for each channel as discussed for the EGRID filter scheme in the previous section.

Areas not
covered by
channels

We propose two distinct methods to deal with locations that are not covered by a channel at all. A simple solution is to define a distinct channel used in that case, hereafter referred to as c_{other} . Obviously, this greatly reduces the precision of event brokering for clients outside the area covered by attraction points. Additionally, the area covered by c_{other} cannot be described as a simple MBR or circle. Consequently, either the client or the broker has to check periodically whether the client's current location is still not covered by another channel. The former requires local knowledge about the areas covered by all channels, while the latter requires periodical updates of a client's location. Given that we deploy channel-based schemes to reduce the computational overhead at the broker, we use the client-side approach in ATTRACT. However, periodic location updates can be enabled for ATTRACT with the self-transition $T^\bullet : f(\text{ATTRACT}) = \text{enable_updates} \leftarrow \text{true}$, as utilized in conjunction with gateway-based event dissemination presented in Section 4.4. The concept of a self-transition is detailed in the following section.

Another way of dealing with locations that are not covered by a channel is the utilization of parametric subscriptions in this case. Therefore, clients that are not covered by a channel report their current location periodically and the broker filters events according to the procedure defined for RADIAL. Which strategy to use depends on the scenario at hand, especially with respect to the density and expected radii of attraction points. Within this work, we use the combination of ATTRACT with the parametric scheme RADIAL, referred to as MULTI, to compare the performance of a single adaptive filter scheme against that of a transition-enabled system. The transition-enabled system realizes the same behavior by executing the respective partial transitions between ATTRACT and RADIAL whenever a client enters or leaves the area covered by an attraction point. This is discussed and evaluated in Section 6.2.4.

4.2.5 Mechanism Reconfiguration through Self-Transitions

Most of the aforementioned filter schemes can be adapted to slight changes in the environmental conditions by altering configuration parameters. Within `STE`, for example, the extension of the envelope can be controlled with the parameter α , as discussed before. For `GRID` and `EGRID`, the grid size could be altered. Within `BYPASS.KOM`, we want to leverage the potential of such self-adaptation mechanisms in addition to the execution of mechanism transitions.

Parameter changes are modeled as *self-transitions*. A self-transition $T^\bullet : f(A)$ is used to alter the state of a mechanism A by applying a custom function $f(\cdot)$ to the currently active mechanism instance. Within `BYPASS.KOM`, we realize the adaptation of filter schemes as self-transition at the broker component of the respective filter scheme. During a self-transition, the respective configuration parameter of the running mechanism instance is updated. The generic realization of self-transitions within the `SIMONSTRATOR.KOM` platform is discussed in Chapter 5.

Self-transitions for parameter changes

In the space-time envelope filter scheme `STE`, the current value of α is only required at the broker component. Clients just report their location and movement vector and all calculations of the resulting envelope are done by the broker. However, for the grid-based schemes `GRID` and `EGRID`, updating the grid size leads to an updated channel assignment. Therefore, as a consequence of self-transitions on the broker, the respective filter schemes utilize their custom context update protocol to inform clients of the newly assigned channels.

It is important to note that the self-transition concept is designed such that it utilizes a scheme's context update protocol. This is due to the observation that most existing mechanisms are already designed with some key properties such as self-healing and self-adaptation in mind. As is the case for the general concept of a transition, we want to utilize such existing properties whenever possible.

Relying on the context update protocol

TYPE	NOTATION
Total transition (Section 4.2.3)	$T : A \rightarrow B$
Partial transition (Section 4.2.3)	$T^{\parallel}(u) : A \rightarrow B$
Self-transition (Section 4.2.5)	$T^\bullet : f(A)$

Table 1: Transition types and notations.

As discussed, the encapsulation of filter schemes as mechanisms in `BYPASS.KOM` and the utilization of a common subscription model enables the execution of transitions between different such schemes. To ensure that the performance of the publish/subscribe system is not degraded while transitions are being executed, we proposed mechanisms for state transfer. Furthermore, we explicitly addressed the potential of including application-specific knowledge and limiting transitions to a specific group of clients by means of partial transitions. We described the integration of distinct classes of filter schemes into `BYPASS.KOM`, enabling us to evaluate the impact of

transitions under unified conditions, as later discussed in Chapter 6. We introduced self-transitions as a generic concept to reconfigure and adapt a mechanism, allowing more fine-grained control over a mechanism’s behavior. Table 1 provides an overview over the transition types introduced in the previous sections. They are further generalized in the SIMONSTRATOR.KOM platform, as later discussed in Chapter 5.

Up to now, we addressed the challenge of adaptivity in location-based filtering. In the following section, we extend the discussion to event dissemination mechanisms and their transitions, addressing adaptive locality-aware event dissemination.

4.3 TRANSITIONS BETWEEN EVENT DISSEMINATION MECHANISMS

In addition to location-based filtering, we aim to achieve locality-aware event dissemination to exploit the locality in user interaction within the communication system. Figure 14 shows how events are processed by the locality-aware dissemination mechanisms of BYPASS.KOM at the client and the broker. The respective functionality depends on the rules applied at the client for instances of (?), as detailed later in this section. To address the challenge of efficiently utilizing a mobile device’s capabilities in terms of direct local communication, we study three distinct approaches to locality-aware event dissemination within BYPASS.KOM: *i*) direct local dissemination initiated by publishers (Section 4.3.1), *ii*) hybrid local and infrastructure-based dissemination (Section 4.3.2), and *iii*) locality-aware hybrid event dissemination using gateways (Section 4.3.4).

Locality-aware dissemination

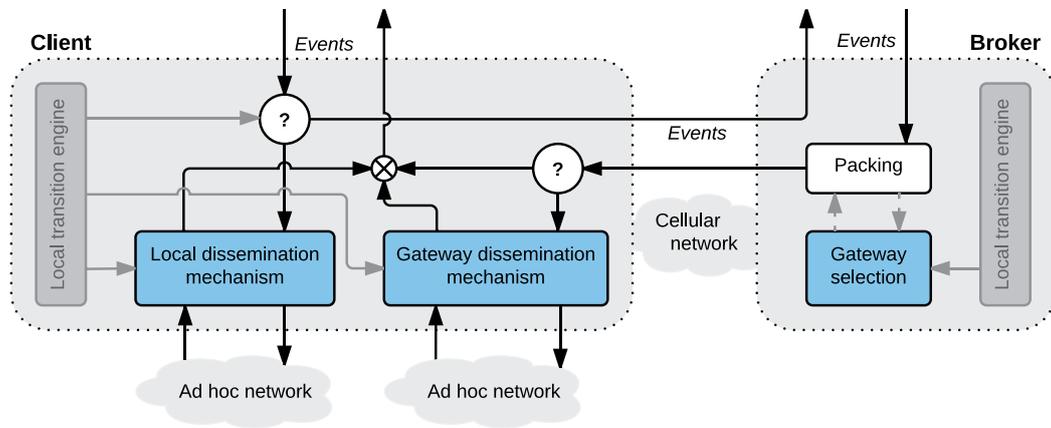


Figure 14: Event dissemination components in BYPASS.KOM.

Locality-aware dissemination does not need to involve a central broker at all but can rely purely on a MANET to distribute and filter events instead. However, pure local dissemination is not feasible if global state is to be maintained at a central entity and for a globally distributed application. It can still be utilized to disseminate events that increase the perceived service quality without affecting global state, as presented in [147], or for applications that operate only within a spatially limited scope [149].

Consequently, for our scenario of location-based mobile social applications, we focus on infrastructure-based and hybrid event dissemination mechanisms, as discussed in Section 4.3.2. Given the research challenge of efficient utilization of the cellular connection, we further introduce mechanisms to reduce the load on the cellular infrastructure during the dissemination of events. To this end, we proposed a stateless approach to gateway selection in [142], detailed in Section 4.3.4. Our approach utilizes different selection algorithms and switches between gateways at no additional cost. This results in a highly adaptive and efficient locality-aware event dissemination in dynamic scenarios.

4.3.1 Local Ad Hoc Event Dissemination

If events are known to be relevant within proximity of their producer, local ad hoc dissemination mechanisms can be utilized to forward them to interested subscribers in proximity. Thereby, locality-aware dissemination is achieved, as events do not need to be processed at a centralized broker entity. Instead, clients themselves filter incoming events against their own subscriptions. Within our application scenario, subscriptions strongly depend on the subscriber's location and are, thus, subject to frequent updates. Therefore, we focus on the efficient dissemination of events while keeping subscriptions locally at each client.

In scenarios with more static subscriptions or less dynamic client mobility, one could also disseminate subscriptions instead. In this case, clients creating an event can already determine the set of interested subscribers and limit event delivery accordingly, as proposed in [72]. A combination of both approaches would involve rendezvous-based matching of subscriptions, as utilized in [45, 199]. We studied the respective characteristics in [147, 149], confirming that pure dissemination of events outperforms structured or rendezvous-based approaches in a dynamic scenario.

Although pure ad hoc dissemination of events cannot scale to a global scenario, we discuss dissemination mechanisms in `BYPASS.COM` as the foundation for our hybrid approaches detailed in Section 4.3.2. Dissemination mechanisms offer a simple API to send an event to nearby clients:

```
void disseminate_locally(event, targets)
```

The second argument, `targets`, can be used to provide a set of target subscribers of the current event to the dissemination mechanism. The target subscribers are not known to a client when publishing an event, and they are not required for broadcast-based dissemination mechanisms. However, when acting as a gateway as discussed in Section 4.3.4, the cloud-based broker can optionally include the respective information to utilize unicast-based dissemination.

To be informed of incoming events, the application (or other components within `BYPASS.COM`) can register as a listener:

```
void add_listener(listener)
```

Each time an event is received by the respective dissemination mechanism, the corresponding method is invoked on all listeners:

```
listener.event_arrived(event)
```

Additional methods are provided to remove listeners, if they are no longer utilized. To enable parallel operation of multiple instances of transition-enabled dissemination mechanisms, the port used for listening for incoming messages can be configured upon initialization. This is later utilized to distinguish between dissemination mechanisms used for direct dissemination of events and those used by gateways, as discussed in Section 4.4.

In the following, we briefly discuss the ad hoc event dissemination mechanisms that are available within BYPASS.KOM. As discussed in our analysis of the state of the art, we do not aim at developing new dissemination mechanisms. Instead, existing mechanisms are integrated into BYPASS.KOM to study their potential transitions.

One-Hop Dissemination via Unicast and Broadcast

As a basic primitive for local dissemination, unicast and broadcast via ad hoc network connections are supported within BYPASS.KOM. These mechanisms do not require any coordination between mobile clients prior to data transmission. However, assuming that the MANET does not utilize a network routing protocol, messages can only reach recipients within direct transmission range of the sender. While unicast messages are only processed by the intended receiver, broadcast messages are processed by all clients within transmission range.

Given the dynamic nature of the underlying network, we utilize UDP as basic primitive for all message transmissions. While broadcasts cannot be realized via TCP, direct unicasts in a dynamic environment would also suffer from TCP's bandwidth-probing approach and frequent disconnects due to client mobility, as shown by Fu et al. in [50]. Consequently, all dissemination protocols integrated within BYPASS.KOM utilize UDP broadcasts and unicasts as their basic messaging primitives. This way, the mechanisms can be utilized on all IP-based network stacks for direct ad hoc communication, including Wi-Fi and Bluetooth. One-hop communication can be utilized by selecting suitable gateways that inject information into their neighborhood. This is discussed in Section 4.3.4.

Multi-Hop Event Dissemination

To disseminate information in larger ad hoc networks, multi-hop communication is necessary. Based on our survey of ad hoc dissemination mechanisms in Section 3.1.2, we integrate four multi-hop dissemination protocols as transition-enabled mechanisms into BYPASS.KOM. These mechanisms vary in their complexity in terms of state management and attempted re-transmissions of messages. In general, we aim to cover the full spectrum of low-complexity, low overhead, and potentially lower reliability up to high complexity, high overhead, and higher reliability.

FLOODING. With FLOODING, each event is re-broadcast once on reception by each client. To detect and discard duplicates, each message is tagged with a Universally Unique Identifier (UUID) on creation. FLOODING causes significant load

on the MANET, especially in densely populated areas. However, the aggressive dissemination strategy leads to high reliability as long as the network is not overloaded, as evaluated in Section A.3. In addition, the mechanism is robust against mobility, as no topology or state information about a client's neighborhood needs to be maintained.

PROBABILISTIC GOSSIP. To reduce the load on the network while at the same time maintain the robustness properties of FLOODING, messages in GOSSIP are forwarded only with a given probability p . For $p = 1$, the mechanism behaves exactly like FLOODING, given that in GOSSIP we also utilize UUIDs to discard duplicate messages. There exist countless variations of gossip-based mechanisms in the literature, as discussed in Section 3.1.2. We chose a simple probabilistic approach given that we want to maintain low delivery delays. Still, by executing self-transitions as defined in Section 4.2.5, we can adapt the respective forwarding probability during runtime.

HYPERGOSSIPING. Proposed by Khelil et al. in [86], HYPERG is a self-adaptive dissemination mechanism based on GOSSIP. In contrast to GOSSIP, HYPERG adapts the probability of forwarding a message based on the current client density within proximity. To this end, a client within HYPERG periodically broadcasts a **Hello** message, allowing other clients to infer their approximate neighborhood based on the received **Hello** messages. Additionally, HYPERG utilizes a random hesitation time before rebroadcasting a message. If a configured amount of copies of the respective message are received during that hesitation time, the message is simply discarded.

PLAN-B. Holzer et al. propose PLAN-B in [68]. In contrast to the aforementioned dissemination mechanisms, PLAN-B relies on location information to adapt the forwarding behavior of clients. When broadcasting a message, the sender piggybacks its physical location to the message. Receivers calculate the distance between the sender and their own location, based on which they determine a hesitation time. The hesitation time is chosen such that clients that are located farther away from the sender send the message first. Thereby, PLAN-B intends to cover a large area with only a minimum number of forwarders. In addition, clients in PLAN-B aggregate incoming messages within a configurable time frame before forwarding them as one larger packet. According to [68], this further increases the achievable throughput on the wireless medium.

In all of the aforementioned mechanisms, the spread of events across the MANET can be restricted by introducing hop counters and a Time to Live (TTL) for messages. However, this would limit the achievable completeness if events are to be disseminated to more remote areas of the network. Given our focus on location-based publish/subscribe, we instead apply the concept of geofencing to reduce load on the network as discussed in the following section.

Utilizing Geofencing for Location-based Events

In our application scenario, location-based subscriptions always cover the physical proximity of a mobile client. Therefore, events published to a specific location do not necessarily need to be distributed across the whole MANET. Given that the size of the AoI is known (or at least has a well-known upper bound) the dissemination mechanism can utilize this knowledge to limit propagation of events to uninterested clients. This process is referred to as *geofencing*, meaning that clients check whether received messages are still relevant at their current locations before forwarding them.

In addition to basic geofencing, dedicated *geocasting* protocols for MANETs have been proposed [112]. They focus on forwarding messages to a geographic location or target area instead of a well-known IP address. We conducted a study on geocasting protocols for location-based services in an earlier work [59], showing the benefits of exploiting data and request locality within ad hoc dissemination mechanisms.

*Geofencing
for local
events*

In BYPASS.KOM, we distinguish between local events and generic location-based events. This allows us to apply the concept of geofencing to our local dissemination mechanisms. A local event is only forwarded by a client if it matches one of the client's subscriptions. Assuming similar AoIs, events are only distributed locally within reasonably close proximity from their source.

However, this mechanism cannot be applied to generic location-based events. Here, more sophisticated geocasting protocols would need to be applied to reduce the spread of messages to areas where the information is not required. However, as we later discuss in the evaluation of BYPASS.KOM, distributing events over large distances in a pure ad hoc fashion is not feasible due to intermittent message loss and partitions of the network. Instead, we utilize transitions between local dissemination mechanisms as discussed in the following section. Executing transitions between dissemination mechanisms allows us to form islands of clients where events are distributed in an ad hoc fashion, as proposed in [147] and detailed in Section 4.4. With that, the concept of geofencing is also applied to generic location-based events.

4.3.2 Cloud-based Hybrid Event Dissemination

As proposed in our previous works [147], we combine direct ad hoc dissemination of events with event dissemination via a cloud-based backend. The resulting hybrid system achieves low delivery delays for events that are distributed within vicinity of a client via ad hoc communication. At the same time, reliable event delivery can be realized via the cloud-based backend. We distinguish between two strategies, as illustrated in Figure 15:

CELLULAR OFFLOADING. Clients only send events to the cloud which are not destined at a location within close proximity. From there on, the event is distributed by the cloud. All locally relevant events are only disseminated via an ad hoc dissemination mechanism. While this reduces the dissemination delay and lowers the utilization of the cellular connection, the successful delivery of local events cannot be guaranteed. Additionally, as the cloud does not receive

all events at least once, a consistent global state cannot be guaranteed. This strategy is utilized by the client on the left side in Figure 15.

CLOUD AUGMENTATION. Clients send their events to the cloud where they are processed and then forwarded to all interested clients. Locally relevant events are additionally sent via an ad hoc dissemination mechanism with geofencing enabled, as discussed in the previous sections. Assuming a reliable cellular connection, as discussed beforehand, this strategy ensures event delivery. For local events, this substantially lowers the delay, as events are ideally received directly via the ad hoc dissemination mechanism. While this strategy does not lower the load on the cellular network, it guarantees that each event is delivered to interested clients at least once. The respective configuration is illustrated in Figure 15 for the client on the right.

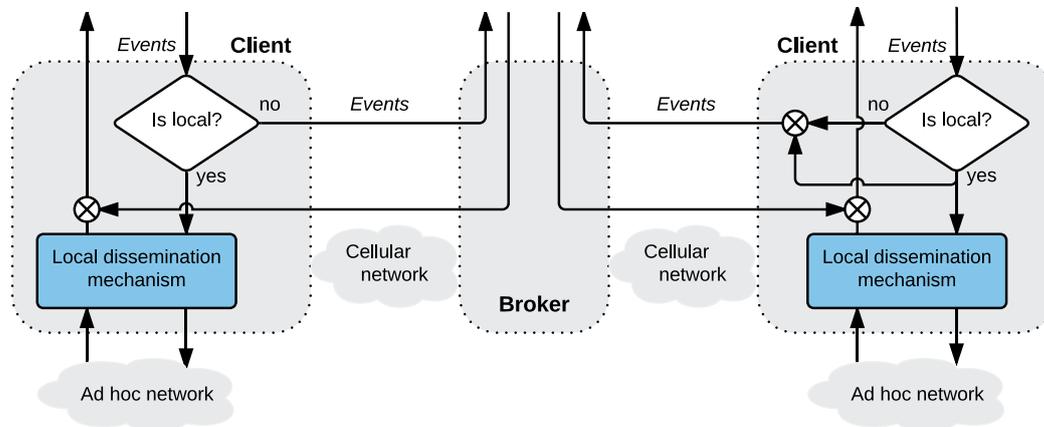


Figure 15: Hybrid dissemination strategies: offloading (left) and augmentation (right).

Combining the hybrid system design with transition-enabled dissemination mechanisms allows for central coordination of mechanism transitions, as presented in [147]. Furthermore, information available at the cloud-based broker helps in selecting suitable dissemination mechanisms. In the following section, we discuss how they are executed within BYPASS.KOM with and without central coordination.

4.3.3 Executing Transitions between Local Dissemination Mechanisms

We consider two general models for the execution of transitions between different dissemination mechanisms: *i)* central coordination and notification, and *ii)* fully decentralized execution. The former requires a central entity, given by the cloud-based broker in hybrid dissemination models as originally proposed in [147]. The latter is required in fully decentralized settings, as discussed in [149] for the scenario of a large-scale music festival.

A centrally coordinated transition between two dissemination mechanisms behaves just like a total transition between different filter schemes, as discussed in

*Centrally
coordinated
execution*

Section 4.2.3. The coordinator (in our case: the broker) notifies each affected client of the transition, which is then executed locally. As for filter schemes, we refer to the respective transition between dissemination mechanism A and B as $T : A \rightarrow B$. In contrast to the transition between filter schemes, the broker itself does not execute a transition, as it does not utilize any ad hoc dissemination mechanism. As soon as a client receives the respective trigger, it executes the transition by interacting with the lifecycle of the respective mechanism proxy, as discussed in Section 5.2. We do not transfer any state information other than the set of currently registered listeners for incoming messages. This is due to the fact that the dissemination mechanisms utilized in our work do not construct complex topologies, as reasoned in Section 3.1.2.

In cases where we do not have a central entity, the trigger for a transition needs to be distributed to all clients via the ad hoc dissemination mechanism [149]. In that case, a decision might not reach all clients due to message loss or partitions within the network, leading to multiple dissemination mechanisms being used concurrently within the network. The same situation can arise for centrally coordinated transitions if the cellular connection is assumed to be unreliable or if the transmission delay varies substantially among clients.

*Self-Healing
Mechanism*

To address this issue, we propose a self-healing mechanism for transition-enabled dissemination in BYPASS.KOM. This mechanism addresses two issues: *i)* detecting a conflict and *ii)* resolving the conflict. To be able to detect a conflict, all dissemination mechanisms within one transition-enabled component utilize the same port to listen for incoming messages. Additionally, each outgoing message contains a flag f_M identifying the mechanism M that created the message. Whenever a client receives a message that contains a flag which does not match the locally utilized dissemination mechanism, a conflict is detected. To provide this functionality in a mechanism-independent way, all outgoing and incoming messages are processed by a message proxy as already introduced for filter schemes in Section 4.2.2.

In order to resolve a detected conflict, additional information is required to decide which mechanism is to be used. Within our scenario, we assume that transitions are initiated by a single entity and not by individual clients. Therefore, we utilize a Lamport timestamp [100] by simply incrementing a counter whenever a transition is initiated. The value of this counter is contained within the respective transition execution plan distributed to clients. Clients simply piggyback their current value to outgoing messages. When receiving a message, clients simply compare their local value of the counter against the received value. In case the received value is greater than the local value, the transition $T : M_\emptyset \rightarrow M$ is being executed, with M_\emptyset serving as generic placeholder matching any currently utilized mechanism, as further discussed in Section 4.4. The target mechanism M is identified by the flag f_M contained in the incoming message.

This self-healing mechanism maintains proper operation of the overall dissemination procedure by ensuring that clients eventually use the same mechanism. However, it cannot be applied to self-transitions (used to adapt parameters of a mechanism) or if transition decisions originate from multiple clients as result of a decentralized decision process. In such cases, more elaborate coordination strategies need

to be deployed, as discussed in Section 5.2.3. We also consider using the self-healing mechanism as a lightweight alternative to global coordination to spread a transition decision within a group of clients, as later discussed in Section 4.4.

To further lower the load on the cellular infrastructure and, consequently, the associated cost for clients, we study the potential of gateways for locality-aware event dissemination in the following.

4.3.4 Locality-aware Dissemination with Stateless Gateways

In each of the aforementioned hybrid dissemination strategies, the cloud-based broker still has to forward location-based events to a potentially large group of interested clients. Given the nature of location-based applications, these interested clients are usually within proximity of each other. Therefore, instead of delivering the event to each client, the broker can select a number of clients as gateways and instruct them to forward the event to their neighbors utilizing an ad hoc dissemination mechanism, as illustrated in Figure 16. Therefore, the subscribers to an event are processed according to the currently utilized gateway selection algorithm as discussed in the following. The list of gateways and their assigned clients is then used to pack the outgoing events. If clients received a packed message with the respective instructions, it acts as gateway, as discussed in [142]. Besides notifying the local filter scheme, a transition-enabled local dissemination mechanism is used to distribute the event to assigned clients nearby.

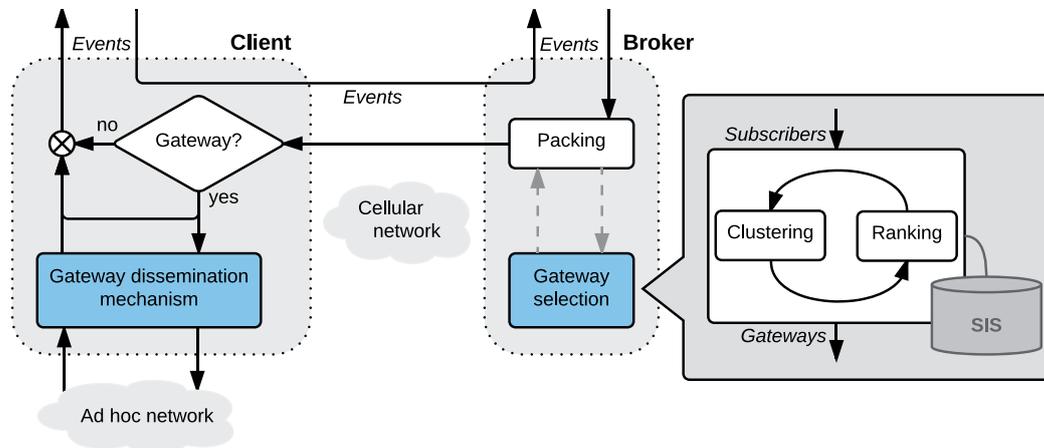


Figure 16: Gateway-based locality-aware dissemination of events.

This reduces the number of outgoing messages at the broker, lowering the load on the cellular network especially for densely populated regions. As discussed in [156], gateway selection algorithms can target different optimization goals, for example: *i)* fair distribution of traffic and energy consumption of clients, *ii)* maximum coverage with the minimal number of gateways, or *iii)* longevity of selected gateways.

*Stateless
gateways*

We studied the impact of different gateway selection algorithms within the process of event dissemination in [142]. Given the dynamic nature of our scenario, we designed the role of a gateway in BYPASS.KOM to be stateless, with the broker entity managing state associated to the respective gateway selection algorithm. As a consequence, frequently switching gateways does not infer any additional cost in terms of messaging overhead. This is in stark contrast to existing approaches for gateway selection, as discussed in [142]. As gateways do not manage any state associated to the process of gateway selection, we can realize transitions between different gateway selection algorithms by exchanging the respective algorithm at the broker. Additionally, state information gathered and maintained by the broker can be transformed and transferred between different gateway selection algorithms.

*Standalone
gateway
selection
algorithms*

More sophisticated gateway selection algorithms require additional information about mobile clients, such as the current battery state. Therefore, instead of integrating them into BYPASS.KOM, gateway selection algorithms are realized as standalone modules in a joint work with Nils Richerzhagen [156]. These modules retrieve all required information via the State Information System (SIS), which is later discussed in detail in Section 5.2.4. Part of this information can be fed into the SIS directly at the broker without additional communication cost; for example, the last known location of a mobile client. Being part of the SIMONSTRATOR.KOM platform, the SIS and its functionality is discussed in detail in Section 5.2.4.

We only consider gateway selection algorithms that operate solely on the last known location of a client, given that this information is available at the broker. Additional selection algorithms discussed in [156] could be utilized if the required state (e. g., battery status) is provided by an additional monitoring mechanism. Following the methodology proposed in [156], we distinguish between *i*) a deterministic gateway selection followed by a clustering step to assign clients to gateways (DKC), *ii*) a stochastic gateway selection followed by a clustering step (SKC), and *iii*) an initial clustering step followed by a stochastic gateway selection within each cluster (CS).

*Deterministic
selection*

In DKC, k gateways are selected based on a deterministic ranking function. The ranking function assigns a weight to each client. The k clients with the highest weight are selected as gateways. Within our work, we utilize a deterministic variant of the LEACH algorithm proposed in [64] as ranking function. Consequently, we refer to this selection algorithm as DKC-LEACH.

*Stochastic
selection*

The stochastic gateway selection procedure SKC also selects k gateways. However, in contrast to DKC, k is treated as an expected value rather than a strict limit. Consequently, the stochastic strategy might assign more than (or less than) k gateways. We again utilize the LEACH algorithm as ranking function, this time including its random cluster-head rotation mechanism as defined in [64]. The resulting stochastic selection strategy is termed SKC-LEACH.

*Cluster-based
selection*

The aforementioned algorithms select gateways and subsequently assign clients to their nearest gateway according to a cluster function. The third strategy, CS, starts by clustering clients before selecting a single gateway within each cluster. We use the WCA algorithm [18] to determine a gateway within a cluster. WCA takes a client's location and its current velocity into account, as discussed in detail in [18]. The cluster-

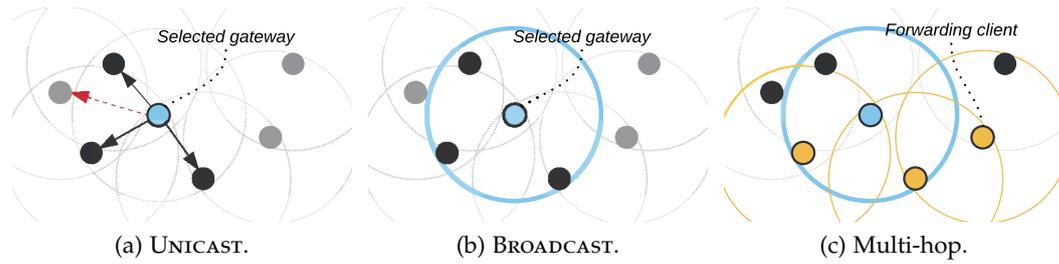


Figure 17: Single-hop dissemination (UNICAST, BROADCAST) and multi-hop dissemination.

based gateway selection strategy relying on WCA is hereafter referred to as CS-WCA. Please refer to [156] for a detailed discussion of alternative selection algorithms and other ranking functions.

Regardless of the currently utilized algorithm, the gateway selection module is accessed via the API method `get_gateways(out_of, k)`, with `k` corresponding to the (expected) number of gateways as previously discussed. The variable `out_of` contains a list of client identifiers out of which gateways are to be selected. Based on the identifiers, additional state information (such as the location of the client) is retrieved by the respective selection algorithm using the State Information System. This is further discussed in Section 5.2.4. The method returns a mapping of selected gateways and their associated clients. This mapping is then utilized to bundle the corresponding events at the broker and forward them to the selected gateways, as discussed in [142]. Thereby, transition-enabled mechanisms for local event dissemination can be utilized in a hybrid dissemination mode.

*API for
gateway
selection*

Within BYPASS.KOM, we need to consider the dependencies between a local dissemination mechanism and the chosen gateway selection algorithm, as illustrated in Figure 17. Considering one-hop dissemination, as studied in [142], gateways need to be selected such that all interested clients are within reach of direct communication. Combined with multi-hop dissemination mechanisms as discussed previously, a single gateway might be sufficient to reach all interested clients in a target area. However, by selecting additional gateways that inject the event into the local ad hoc dissemination mechanism, the reliability of the overall dissemination process can be increased. We study the interdependencies between gateway selection and local dissemination within the scope of coexisting transition-enabled mechanisms, as discussed in the following section.

4.4 COEXISTENCE OF TRANSITION-ENABLED MECHANISMS

To combine location-based filtering with locality-aware event dissemination, the respective mechanisms need to coexist in a publish/subscribe system. Additionally, transitions between these coexisting mechanisms need to be executed to adapt the system to application-specific client mobility and workload characteristics. In the following, we discuss the combined utilization of mechanisms for location-based filtering and locality-aware dissemination of events in BYPASS.KOM. We propose to

adapt the choice of mechanisms to deal with application-specific client mobility by executing transitions near application-defined attraction points.

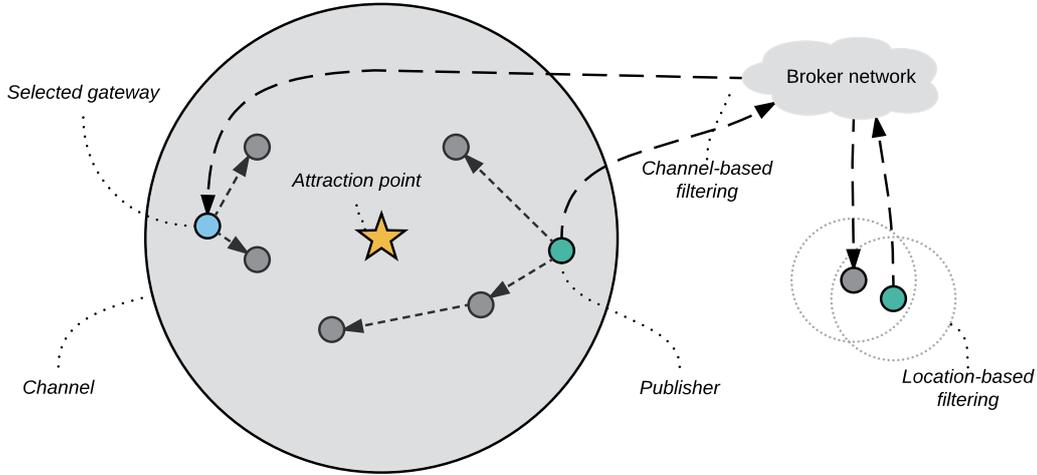


Figure 18: Application-specific utilization of coexisting mechanisms.

To this end, we assume that the locations and radii of attraction points are provided by the application, as previously discussed for the channel-based filter scheme ATTRACT in Section 4.2.4. As illustrated in Figure 18, we consider two application-specific scenarios that require coexisting transition-enabled mechanisms. The scenarios resembles the hybrid dissemination strategies discussed in Section 4.3.2. However, the effects are limited to the area around attraction points and further combined with transitions between the utilized location-based filter schemes.

CX-ADHOC. Ad hoc dissemination of local events is utilized in the area covered by an attraction point to reduce the delay of event delivery. In addition, a channel-based filter scheme is used for all clients in the area defined by an attraction point to avoid sending frequent context update messages. Clients not within proximity of an attraction point use a parametric filter scheme. Furthermore, they do not utilize an ad hoc dissemination mechanism.

CX-GATEWAY. We utilize local dissemination mechanisms and filter schemes as discussed for CX-ADHOC. However, we additionally deploy stateless gateways to distribute events targeted at a location within a channel to all clients within vicinity of the attraction point. Instead of sending the event to each client individually, gateways combined with local dissemination are utilized to offload the cellular connection around attraction points.

In both cases, clients utilize a parametric filter scheme M_{pm} until they approach an attraction point. Within a circular area around attraction points, a channel-based filter scheme M_{ch} is used instead. The respective transition is initiated as soon as a client enters or leaves the area covered by an attraction point. As the transition only affects the client u entering (or leaving) the area, it is modeled as a partial transition $T^{\parallel}(u)$:

$M_{pm} \rightarrow M_{ch}$ (or $T^{\parallel}(u) : M_{ch} \rightarrow M_{pm}$, respectively) as discussed in Section 4.2.3. Consequently, the client's state is migrated from one scheme's broker component to the other. In addition to switching the filter scheme, the client is instructed to activate a local ad hoc dissemination mechanism that is utilized whenever the client issues a local event. The dissemination mechanism is implicitly subject to geofencing, as ad hoc dissemination is deactivated once a client leaves the area of an attraction point.

For the case of *CX-GATEWAY*, we additionally utilize stateless gateways as previously discussed in Section 4.3.4. These gateways are utilized to disseminate events to all subscribers within a given channel, i. e., the clients in the area defined by an attraction point. The utilization of gateways requires a local dissemination scheme on both, gateways and the intended receivers. Therefore, in *CX-GATEWAY*, clients operate two instances of a transition-enabled local dissemination mechanisms. One instance is used for the ad hoc dissemination of local events, as previously discussed. The second instance is used by client acting as gateways to forward events received by the broker to nearby clients. By utilizing separate transition-enabled dissemination mechanisms, we can adapt the mechanisms independent of each other. This is motivated by the evaluation results later discussed in Section 6.4.2 and further supported by our findings in [142]. We show that stateless gateways work efficiently with one-hop dissemination mechanisms like *UNICAST* and *BROADCAST* introduced in Section 4.3.1, whereas the direct dissemination of local events requires a multi-hop dissemination mechanism to reach all interested clients.

We model the aforementioned transitions based on the generic concept of *execution plans* provided by the *SIMONSTRATOR.KOM* platform. Therefore, we present the resulting execution plans and concrete transitions when discussing our prototype of *BYPASS.KOM* in Section 5.3.2. We further discuss the adaptation of the respective execution plans to the currently observed client density around an attraction point.

With *BYPASS.KOM*, we introduce a framework that allows us to design and execute transitions between distinct mechanisms for location-based filtering and locality-aware event dissemination, respectively. We proposed distinct transition types and a common methodology for their execution. We focus on two perspectives: *i*) individual mechanisms and their transitions and *ii*) coexisting mechanisms with interdependent transitions. While *BYPASS.KOM* targets the domain of location-based publish/subscribe, we generalize core concepts and mechanisms in our second contribution, the *SIMONSTRATOR.KOM* platform. In the following chapter, we discuss the *SIMONSTRATOR.KOM* platform as an environment for the design and evaluation of transition-enabled communication systems.

SIMONSTRATOR.KOM: PLATFORM FOR THE DESIGN AND EVALUATION OF MECHANISM TRANSITIONS

INSIGHTS obtained through domain-specific studies of mechanism transitions (as done with BYPASS.KOM) need to be consolidated and generalized towards a common transition methodology. This requires intense collaboration among researchers for the definition and later adoption of common design practices. With the SIMONSTRATOR.KOM platform [148, 155] as our second contribution, we support this process from the design of transition-enabled mechanisms to their evaluation in heterogeneous scenarios.¹ More specifically, the SIMONSTRATOR.KOM platform serves the following goals: *i)* foster collaboration among researchers by supporting rapid prototyping and loose coupling among specific components, *ii)* consolidate generalized design concepts for transition-enabled communication systems and their coordination, and *iii)* support evaluations based on simulations and prototypical deployments.

We provide an overview of the platform in the following section, before discussing the transition-specific contributions in Section 5.2. These contributions deal with the design of transition-enabled mechanisms and their unified coordination through a *transition engine*. Finally, we discuss our prototype realization of BYPASS.KOM within the SIMONSTRATOR.KOM platform, serving as the foundation for the evaluation presented in Chapter 6.

5.1 OVERVIEW OF THE PLATFORM

The Java-based SIMONSTRATOR.KOM platform consists of three main building blocks, as illustrated in Figure 19: *i)* a core *framework* for distributed communication systems, *ii)* a number of *runtime environments* for execution and evaluation, and *iii)* realizations of *specific mechanisms* within the framework. Realizations of mechanisms are usually separated into multiple projects, given that they represent individual researchers' current work. They are designed and implemented using the APIs provided within the core framework. This ensures interoperability between mechanisms and enables their execution in different runtime environments, as detailed in the following. An in-depth description of the platform and its components is provided in [148], albeit not focusing on transition-enabled mechanisms. In the following, we provide a brief summary of the platform as a foundation for the discussion of the transition-specific components and design concepts contributed in this thesis.

¹ The SIMONSTRATOR.KOM platform, its runtime environments and the prototype of BYPASS.KOM described in this thesis are available online at www.simonstrator.com [Accessed March 8th, 2017].

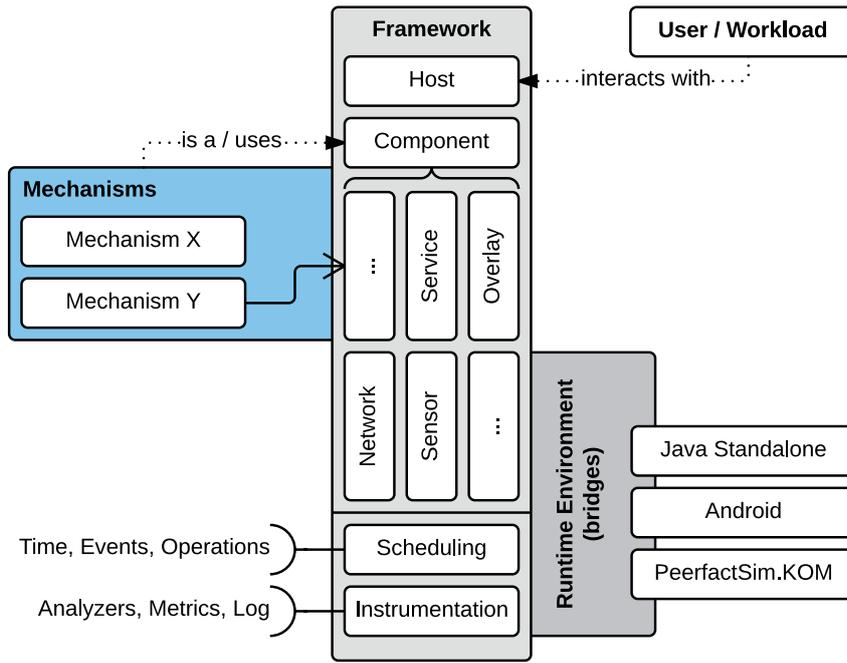


Figure 19: Overview of the SIMONSTRATOR.KOM platform (adapted from [148]).

5.1.1 The Core Framework: Time, Events, and Instrumentation

Global components

The core SIMONSTRATOR.KOM framework consists of mechanism-specific APIs and a set of core components (or *global components*) providing abstractions for *i*) scheduling of events, *ii*) relative time calculations, and *iii*) instrumentation. Abstractions for time and scheduling of events are required to support runtime environments that do not rely on continuous time, such as event-based simulators. Instrumentation is required to assess the performance of a mechanism, both in simulations and in real-world deployments. By using the provided abstractions, one can benefit from an extensive set of tools for collection and post-processing of data. Implementing a mechanism using the provided abstractions ensures that it can be executed on any of the runtime environments available within the SIMONSTRATOR.KOM platform, as later discussed in Section 5.1.3.

Scheduling future events

Scheduling a specific operation at a later point in time is a key functionality that is usually achieved in application threads through the use of sleep cycles. Within the SIMONSTRATOR.KOM, all scheduling is done through the Event wrapper provided by the framework:

```
void Event.schedule_with_delay(delay, event_handler, content, event_type).
```

The time until the event is executed, *delay*, is given in multiples of the time units provided by the Time wrapper, such as Time.MINUTE or Time.SECOND. Once the event is to be executed, the provided *event_handler* is notified by invoking

```
void event_occured(content, event_type).
```

The parameter content allows the user to pass an arbitrary object that is then passed to the respective event handler, as discussed hereafter. Similarly, the optional integer parameter `event_type` is passed to the event handler to enable easy distinction in case of multiple event types being processed by the same handler. For relative time calculations or absolute timestamps, rather than using the system clock, the current timestamp is obtained through the `Time.get_current_time()` method. Combined with the time units provided by the Time wrapper, this ensures identical behavior of all time-related calculations regardless of the utilized runtime environment.

Accessing the current time

Instrumentation aids in debugging during the design phase and in conducting measurements during subsequent evaluations. Within the SIMONSTRATOR.KOM platform, we offer pull- and push-based instrumentation in addition to simple logging functionality, as illustrated in Figure 20. Pull-based instrumentation relies on a simple, predefined interface that enables access to specific internal states of a system via a *metric*. State can be accessed and observed through this interface in a simple and unified fashion, usually through periodic sampling. In contrast, push-based instrumentation allows the system to report specific actions or conditions on occurrence. The respective actions are usually mechanism-specific, requiring a custom instrumentation interface, referred to as *analyzer*.

Instrumentation: metrics and analyzers

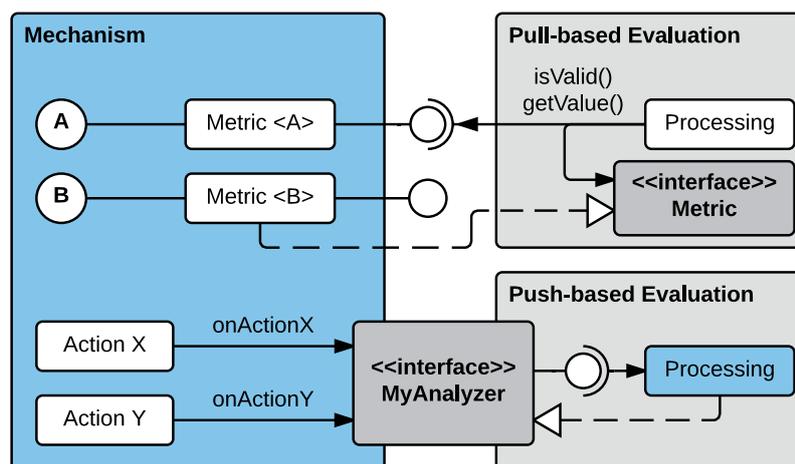


Figure 20: Realization of push- and pull-based instrumentation (adapted from [148]).

Both methods of instrumentation are not expected to alter the internal state of the system that is being evaluated. Consequently, methods defined by a custom analyzer interface cannot return values. This is a fundamental requirement in the process of binding metrics and analyzers to a specific runtime environment. A runtime environment may include multiple analyzers implementing and binding to the same interface. This is transparently taken care of by the runtime environment, which multiplexes the respective method invocations to all analyzer instances. In case that there is no implementation of a specific analyzer available, the runtime environment automatically creates an empty stub.

5.1.2 Component-based Composition of Hosts

In addition to the aforementioned global components, mechanism-specific APIs for so-called *host components* are provided within the framework. A *host* describes an entity of the communication system, including mobile clients and intermediate network devices. The functionality of a host is defined by the set of available and configured host components. The SIMONSTRATOR.KOM framework offers APIs for a number of different protocols and services, including the API for location-based publish/subscribe proposed in Chapter 4. As indicated in Figure 19, additional device-specific capabilities are offered as host components, including access to location sensors like GPS or available network interfaces for cellular and local ad hoc communication.

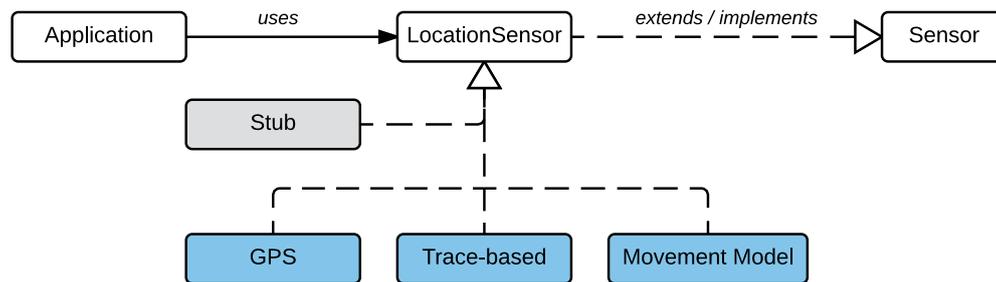


Figure 21: Example of exchangeable component realization (adapted from [148]).

Decoupling of
host
components

Host components can have multiple implementations, often depending on the runtime environment. Which implementation is to be used is configured within the respective runtime, with the individual components communicating with each other only through the APIs defined in the framework.² This enforces decoupling of functional dependencies and the actual realizations of the respective host components. Consequently, stub implementations of specific aspects of a communication system can easily be replaced with current research prototypes. Thereby, researchers can contribute to the big picture without creating too many dependencies on rapidly developing and possibly unstable research prototypes at the same time.

This is illustrated in Figure 21 for the sensor used to determine a host’s current location. The **LocationSensor** extends the generic **Sensor**, and applications interact with the respective interface. The actual realization, however, depends on the runtime environment and its configuration. On Android devices, for example, the device’s actual GPS sensor can be utilized. In simulations, one might choose between a trace-based sensor implementation or a fully-fledged mobility model.

The decoupling within the SIMONSTRATOR.KOM platform has proven to be vital for the Collaborative Research Centre “MAKI” during the development of demonstrators [143, 145, 150, 194]. Before going into detail on our transition-specific contributions to the SIMONSTRATOR.KOM platform, we briefly discuss the runtime environments used for the aforementioned demonstrators and the evaluation of BYPASS.KOM later presented in Chapter 6.

² For a detailed description and examples of configured hosts, the interested reader is referred to [148].

5.1.3 Runtime Environments

The SIMONSTRATOR.KOM platform includes a number of runtime environments for the execution and evaluation of systems developed within the framework. The current release 3.1 of the platform contains the following runtime environments:

PEERFACTSIM.KOM. An environment for event-based simulations relying on an extended version of the PEERFACTSIM.KOM overlay and network simulator [176]. This runtime environment was used for the evaluations presented in this thesis. The respective configuration and the utilized network and mobility models are described in more detail in Chapter 6.

ANDROID. Given our focus on mobile applications, the Android runtime environment executes code on Android smartphones. It provides a basic stub for custom Android applications that can utilize systems and mechanisms implemented within our framework. The smartphone-based demonstrations published in [145, 150, 194] rely on this runtime environment.

JAVA STANDALONE. This environment enables the execution of systems on any computer with a Java runtime installed. This includes testbed environments [9] utilizing a headless mode and demonstration setups with custom user interfaces and visualizations.

It is important to note that the SIMONSTRATOR.KOM platform itself is *not* a simulator or emulator. Instead, it relies on a runtime environment to implement the respective core components for scheduling and time, as discussed previously and detailed in [148]. However, development of the SIMONSTRATOR.KOM platform is closely intertwined with the continued development of PEERFACTSIM.KOM. Consequently, the PEERFACTSIM.KOM runtime is the most feature-complete runtime. Additionally, PEERFACTSIM.KOM has been extended with new models and tool integrations to specifically target the scenario of mobile social applications, as presented in [155, 177]. This includes the integration of OpenStreetMap (OSM) data for map-based navigation as foundation for more sophisticated mobility models as later discussed in Chapter 6.

Based on this brief introduction of the SIMONSTRATOR.KOM platform, we now discuss core contributions supporting the design of transition-enabled systems and the coordination of transitions.

5.2 PLATFORM SUPPORT FOR TRANSITIONS

In our early contributions on transition-enabled mechanisms in the context of P2P video streaming [140, 191], the execution of transitions is realized as part of the streaming system. Consequently, encapsulation of the transition-enabled mechanisms relied on application-specific interfaces that cannot easily be applied to other application domains. Based on the design of BYPASS.KOM and our recent contributions on adaptive monitoring systems [153, 157], we extend and generalize the concept

of transition-enabled mechanisms within the SIMONSTRATOR.KOM platform. The resulting abstractions for *transition-enabled mechanisms* and *mechanism proxies* are introduced in the following section. Utilizing these abstractions, we present an application and system agnostic *transition engine* that executes local transitions in Section 5.2.2. Subsequently, platform support for the coordination and execution of transitions that affect multiple devices based on *execution plans* is discussed in Section 5.2.3. Finally, we discuss the platform support for the monitoring, analysis, and planning steps of the MAPE-cycle in Section 5.2.4.

5.2.1 Transition-enabled Mechanisms and Mechanism Proxies

The execution of transitions is a generic, mechanism-independent concept. Consequently, it has to be decoupled from the mechanism-specific APIs. In the SIMONSTRATOR.KOM platform, this is achieved by *mechanism proxies*, as illustrated in Figure 22 and published in [49]. A mechanism proxy exposes the API of the original mechanism and transparently routes all method invocations to the currently active mechanism instance maintained within the proxy. When executing a transition, the transition engine interacts with the mechanism proxy to switch from one active mechanism to another, potentially executing a custom `AtomicTransition`, as illustrated. This is discussed in more detail in Section 5.2.2. For now, we focus on the mechanism proxies and their utilization.

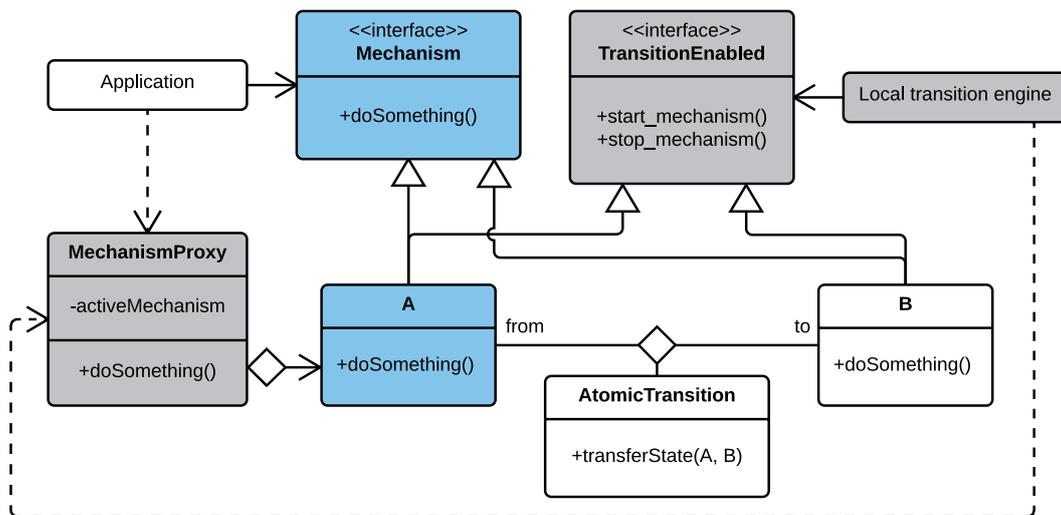


Figure 22: Mechanism proxies within the SIMONSTRATOR.KOM platform.

Proxies for transition-enabled mechanisms are created via the transition engine, a host component within the core framework of the SIMONSTRATOR.KOM. The transition engine offers the following method to create a new proxy for a given mechanism type, described by its application interface C:

```
Proxy create_mechanism_proxy(Class<C> api, default_instance, proxy_name);
```

The default mechanism instance provided when creating the proxy is used as active mechanism until any transitions are executed. The name of the proxy is required to address specific proxies when defining execution plans for transitions, as discussed later. Once the respective proxy is created, it can be utilized by the application just as any customary instance of the mechanism. Existing mechanism proxies can be accessed through the transition engine:

```
Proxy get_proxy(proxy_name, Class<C> api);
```

In case that a proxy is no longer required, the transition engine takes care of a proper shutdown of the currently active component.

If a transition is to be executed, the active mechanism within the respective mechanism proxy is altered or exchanged by the local transition engine. To this end, mechanisms have to implement two lifecycle methods defined by the `TransitionEnabled` interface: `start_mechanism` and `stop_mechanism`. In contrast to our earlier proposal in [49], we substantially simplified the lifecycle within the `SIMONSTRATOR.KOM` platform. In [49], the bootstrapping phase of a mechanism is part of the startup process. However, this definition makes it hard to define upper bounds for the time it takes before a mechanism has finished this process. Given that a transition can only be executed after the target mechanism has started, this leads to non-atomic behavior.

Within `BYPASS.KOM` and the lifecycle and transition model used by the `SIMONSTRATOR.KOM` platform, we follow a more radical approach to maintain atomicity of transitions. A mechanism's startup phase is a purely local operation and must not include communication with other components. Instead, only locally available state obtained during the state transfer phase of a transition can be used. Therefore, the respective method `start_mechanism` returns immediately, enabling atomic execution of transitions as will be discussed in the following section. If additional bootstrapping is required in cases where the state transfer does not suffice, this process is part of the normal operation of the mechanism.

Atomicity of transitions

In some cases, a communication system might require multiple instances of a specific mechanism proxy. This is, for example, the case if gateways are utilized in addition to direct local dissemination within `BYPASS.KOM`, as described in Section 4.4. In this case, one mechanism proxy containing transition-enabled dissemination mechanisms is used for direct dissemination of events, while a second instance is used for gateway-based dissemination. While the respective mechanism proxies can easily be distinguished by their name, the mechanisms themselves need to be designed (or at least configured) such that parallel operation is possible. In case of dissemination mechanisms, they need to use different transport layer ports, for example. This is further discussed in Section 5.3.1 for the prototype of `BYPASS.KOM`.

Multiple mechanism proxies

To support starting and stopping a transition-enabled mechanism, we introduce the *generic mechanism* M_{\emptyset} as source or target mechanism of a transition. Executing a transition from M_{\emptyset} to any other mechanism denotes a startup procedure, whereas a transition to M_{\emptyset} shuts down a mechanism instance. Whenever a transition to M_{\emptyset} is to be executed by a mechanism proxy, the respective proxy transparently provides a stub implementation of the proxy interface. This stub does not offer any functionality other than empty implementations of the methods defined in the interface. Thereby,

The generic mechanism M_{\emptyset}

it is ensured that invocations of the mechanism proxy still result in valid function calls. At the same time, the source mechanism of the transition is correctly terminated via its `stop_mechanism` lifecycle method just as during a normal transition.

If M_\emptyset is specified as source mechanism of a transition (e. g., $T : M_\emptyset \rightarrow A$), it serves as a *wildcard* that matches any currently running mechanism. Thereby, transitions can be executed even if the currently running mechanism is not known to the coordinator, as discussed in the following section.

5.2.2 Transition Engine and Custom Transitions

As briefly mentioned in the previous section, the transition engine takes care of creating and maintaining proxies for transition-enabled mechanisms. Additionally, it is responsible for the local execution of transitions by interacting with the corresponding proxy. To execute a transition $T : A \rightarrow B$ from mechanism A to B, the following method needs to be triggered on the local transition engine:

```
void execute_atomic_transition(proxy_name, Class<C> target_class)
```

Note that the target mechanism's class is required, corresponding to the implementation of B. There is no need to specify A explicitly, as the transition to B is executed irregardless of the currently active mechanism of the proxy. Instead, the `proxy_name` is used to identify the correct mechanism proxy.

Next, the transition engine searches for an `AtomicTransition` object that matches the source and target mechanisms. An atomic transition contains code that is to be executed in order to transfer state between A and B. Custom realizations of an atomic transition can be registered at the transition engine:

```
void register_transition(proxy_name, AtomicTransition<A, B> transition)
```

Within each atomic transition, the method `transfer_state(A, B)` has to be implemented. While the application can only access the methods defined for the proxy upon creation, within `transfer_state(A, B)` one has full access to the individual functions defined by A and B. This, in turn, allows for arbitrary complex mechanism-specific state transformations. However, forcing developers to write explicit code for each potential transition would hinder the integration of new mechanisms and become tedious for larger choices of mechanisms. Therefore, the SIMONSTRATOR.KOM platform supports *i)* fully automated state transfer using annotations and *ii)* type inheritance for atomic transitions, as discussed in the following.

*Automated
state transfer*

Fully automated state transfer relies on Java annotations. To enable automatic state transfer, fields within the respective mechanisms are annotated with `@TransferState`. This is, for example, used in BYPASS.KOM to transfer a reference to a location sensor between filter schemes:

```
@TransferState()  
LocationSensor sensorReference;
```

During a transition, the local transition engine checks for matching annotations at the source and the target mechanism. If, in our example, both mechanisms contain

an annotated field of the type `LocationSensor`, the value of the field is transferred to the new mechanism using the Java Reflection API. This corresponds to a custom `AtomicTransition` implementing the following `transfer_state` method:

```
void transfer_state(source, target) {
    target.sensorReference = source.sensorReference;
}
```

The annotation accepts an optional string value, e.g., `@TransferState("[name]")`. In this case, if the respective state is being transferred to the target mechanism of a transition, the local transition engine searches for a method `set[Name]` that accepts a variable of the respective type. This method is then invoked with the source mechanism's value. Thereby, a mechanism can actively react to transferred state by, for example, initializing an internal data structure. In case of automated state transfer, a *transformation* of the information is not possible. Consequently, source and target mechanism need to use the same state representation (data type). If this is not the case, a custom transition needs to be registered, as previously defined.

When executing a transition, the local transition engine checks, whether a custom transition matches the specified source and target mechanisms. This process supports type inheritance, relying on the respective mechanisms provided by the Java programming language, as illustrated in Figure 23. Here, three mechanisms (A, B, and C) are shown. Each mechanism contains one or more state variables s_i that are annotated with `@TransferState()`. Additionally, the custom transitions $T_1 : A \rightarrow C$ and $T_2 : M_\emptyset \rightarrow C$ have been registered at the local transition engine beforehand.

Type inheritance for atomic transitions

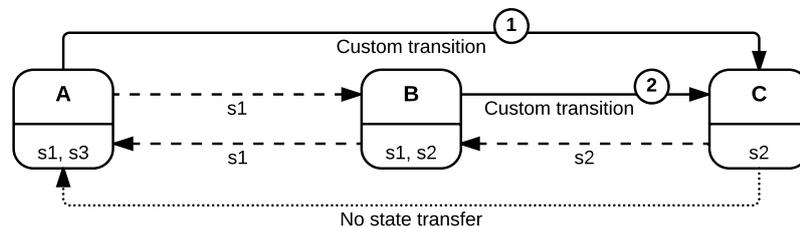


Figure 23: Illustration of state transfer and type inheritance for (custom) transitions.

As shown in the figure, transition ① is executed using the custom implementation provided with T_1 . The custom transition T_2 with M_\emptyset as source mechanism also applies here, but the specification of T_2 is a closer match in terms of the type hierarchy of mechanisms (M_\emptyset serves as parent for all mechanisms). The transition ②, however, is a match for the custom transition T_2 . Consequently, instead of executing an automatically generated transition that would transfer state s_2 , the custom transition is executed. In all other cases, matching state is automatically transferred as the respective state variables were properly annotated. Only for the transition from C to A, no matching state is found and, thus, no state information is transferred. The annotation-based automation of state transfer and the type inheritance for atomic transitions substantially reduce the effort required to integrate new mechanisms.

Irregardless of whether the transition from A to B uses a custom implementation of an `AtomicTransition` or whether it relies on the automated approach, it consists of the following steps, as illustrated in Figure 24:

1. Create a new instance of mechanism B and transfer state from A to B.
2. Trigger `start_mechanism` at B (returns immediately in `BYPASS.KOM`).
3. Switch the active mechanism within the proxy to mechanism B.
4. Trigger `stop_mechanism` at A and discard the instance of mechanism A.

Steps 1 to 3 are considered a single atomic operation, if `start_mechanism` returns immediately, as discussed before.

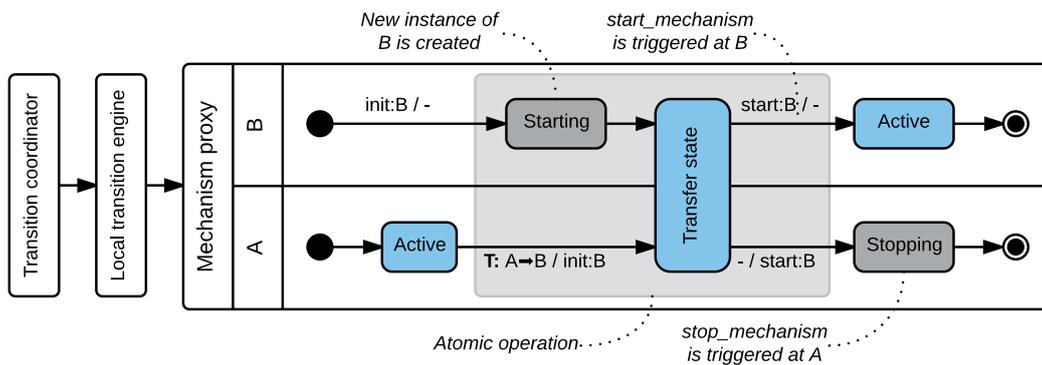


Figure 24: Transition execution in the SIMONSTRATOR.KOM platform.

Self-transitions

Besides the execution of atomic transitions resulting from total or partial transitions within a system, we motivated the concept of self-transitions. Such transitions are used to alter the state of a mechanism without actually switching the mechanism instance. Self-transitions within the SIMONSTRATOR.KOM platform are also executed via the transition engine:

```
void execute_self_transition(proxy_name, SelfTransition<C> transition)
```

Analogous to the functionality of an `AtomicTransition`, a `SelfTransition` realizing $T^\bullet : f(A)$ defines the method `alter_state(A)`, providing a reference to the currently active mechanism instance. Based on the methods available through the instance A, local state can be altered during the self-transition. As for atomic transitions, the SIMONSTRATOR.KOM platform includes annotations for mechanism state supporting the automated creation of matching self-transitions.

By annotating a field with `@MechanismState("[name]")`, the method `set[Name]` is invoked with the updated state variable whenever a self-transition is executed. This is especially useful for self-transitions that simply update a configuration parameter, as discussed in Section 4.2.5. If, for example, $T^\bullet : f(\text{Gossip}) = p \leftarrow 0.4$ is to be executed, and the state variable representing p in the Gossip dissemination mechanism is annotated with `@MechanismState("[Probability]")`, the following `SelfTransition` is automatically created and executed for the current mechanism instance of Gossip:

```
void alter_state(gossip) {
    gossip.setProbability(0.4);
}
```

The transition engine takes care of the local execution of transitions. This is only the last step in a chain of actions required to execute transitions across multiple hosts and multiple different mechanisms.

5.2.3 Coordination of Transitions on Multiple Hosts

Within the SIMONSTRATOR.KOM platform, a *transition coordinator* is responsible for the execution of transitions on multiple hosts. To this end, the transition coordinator supports the creation of an *execution plan*. The execution plan is an ordered sequence of transitions $EP = [T_1, \dots, T_n]$ that are to be executed, including any potentially required self-transitions. An execution plan is executed locally at the respective client; it does not involve any coordination. The execution plan serves as a blueprint that is distributed to all local transition engines that are involved in the global or partial transition. It can store an arbitrary number of *actions*, corresponding to either atomic transitions or self-transitions with associated state information. For each action, the name of the mechanism proxy that will be affected by the action needs to be provided:

*Execution
plan and
actions*

```
plan = TransitionCoordinator.create_execution_plan();
plan.add_action(AtomicTransitionAction(proxy_name, Class<C> target_class));
plan.add_action(SelfTransitionAction(proxy_name, Class<C> target_class,
    state_identifier, state_value));
TransitionCoordinator.execute_plan(affected_hosts, plan);
```

With `execute_plan`, the plan is distributed to and executed on all hosts addressed by `affected_hosts`. Each host is equipped with an instance of a transition coordinator. In BYPASS.KOM, we utilize the cloud-based broker as a centralized entity that coordinates transitions. The transition coordinator component at mobile clients simply executes received plans by invoking the respective actions at the local transition engine. Fully distributed or hierarchically arranged coordinators can also be realized with the proposed structure, given that just the execution plan is predefined. The protocol used between coordinator instances is not defined by the platform and can be of arbitrary complexity, ranging from coordination via the global knowledge within a simulator to fully distributed consensus protocols for a decentralized realization of the MAPE-cycle [36]. In cases where the involved mechanisms cannot benefit from self-healing properties or a reliable network connection, more effort on the coordination side might be required.

5.2.4 Monitoring, Analysis, and Planning

So far, contributions focused on the execution of transitions as the last step of the MAPE-cycle introduced in Section 2.4. In the following, we briefly discuss contributions that support the monitoring, analysis, and planning phase of the cycle.

Providing data about the system's current state is an inevitable prerequisite for the analysis of the system's behavior as well as consequent planning and execution of transitions. Motivated by joint work on monitoring mechanisms [62, 152, 153, 157], we provide a local hub for data from multiple mechanisms within the SIMONSTRATOR.KOM platform. This State Information System, or SIS in short, is a local host component where mechanisms can register themselves as providers or consumers of specific data, based on the concept of monitoring access points proposed in [152]. The SIS interacts with the monitoring mechanism to distribute data across multiple hosts and resolve requests for data that is not locally available. Thereby, it provides data required by mechanisms such as the gateway selection algorithms presented in Section 4.3.4. Additionally, it provides a unified way for mechanisms in a single host to exchange data with each other. If, for example, a dissemination mechanism requires information about the local one-hop neighborhood of a mobile client, it can request that information from the SIS. The SIS tries to resolve the request locally by checking for providers (i. e., local mechanisms) that offer the respective data. When requesting information from the SIS, one can specify whether the SIS may utilize the monitoring mechanism to ask other clients.

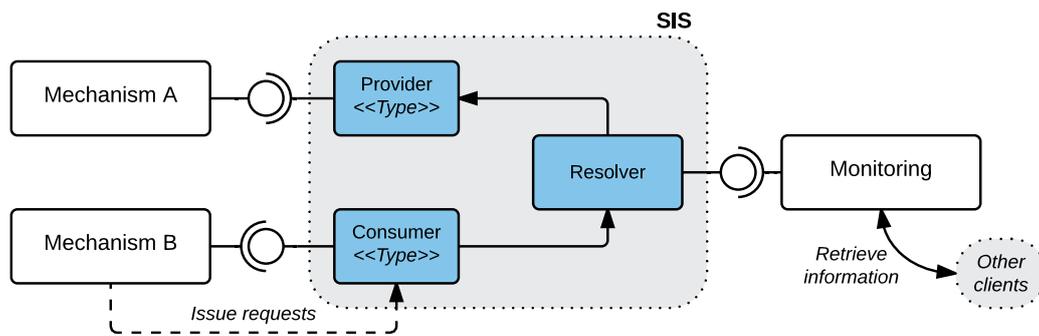


Figure 25: Conceptual overview of the State Information System.

Figure 25 provides a conceptual overview of the SIS and its interactions with information providers, consumers, and the monitoring mechanism. Besides supporting mechanisms during operation, the SIS can also be utilized to bootstrap mechanisms after a transition occurred. In addition to the direct transfer and transformation of state during a transition (as previously discussed), a mechanism can utilize the SIS to request state that is locally available at other mechanisms or resolved using the monitoring mechanism. The benefits resulting from this approach of bootstrapping mechanisms after a transition have been studied in the context of network routing protocols for MANETs in [190].

As is the case for all components within the SIMONSTRATOR.KOM platform, the actual realization of the SIS can differ depending on the runtime environment and its configuration. During early stages of mechanism design and the associated evaluations in a simulation environment, this allows us to rely on global knowledge provided by the simulator to single out other effects. A more realistic monitoring-based

realization of the SIS can be used at any point in time without requiring modifications to the mechanisms utilizing it.

Information gathered by the monitoring mechanism and provided through the SIS is required during the analyzing and planning steps of the MAPE-cycle. We follow a pragmatic approach to analysis and planning, given that these phases of the MAPE-cycle are not in our core focus. In *BYPASS.KOM*, transitions are executed based on predefined rules, as presented in Section 4.4. All transitions are planned and coordinated by the cloud based broker entity based on locally available state information such as the current location or the number of subscribers.

*Analysis and
planning*

Even though the analysis and planning steps are not within the focus of this thesis, other researches can benefit from concrete transition-enabled mechanisms available within the *SIMONSTRATOR.KOM* platform. More elaborated approaches to analysis and planning are explored in collaboration with other researches. This includes the description of transition-enabled systems in variability models to enable automatic derivation of valid system configurations, as done for *BYPASS.KOM* in [189]. Additionally, automated inference of suitable rules for the execution of transitions based on machine learning and genetic programming is studied in [47, 48].

5.3 PROTOTYPE OF BYPASS.KOM

Within this section, we discuss the prototype of *BYPASS.KOM* realized based on the *SIMONSTRATOR.KOM* platform. We focus on the utilization of mechanism proxies, including parallel utilization of multiple instances and discuss how *BYPASS.KOM* is integrated into the MAPE-cycle for the coordination and execution of transitions.

5.3.1 Mechanism Proxies Utilized by Clients and Brokers

As discussed in Chapter 4, *BYPASS.KOM* is focused on two main functional components: filtering of events and local dissemination mechanisms including the utilization of gateways. To this end, each client utilizes five mechanism proxies, as described in the following and illustrated in Figure 26.

LOCAL TRANSPORT. This mechanism proxy provides transition-enabled access to a local transport interface and the underlying physical network technology. Thereby, dissemination mechanisms can operate regardless of the underlying network and transport technology. In *BYPASS.KOM*, the proxy provides mechanisms for UDP unicasts and broadcasts over Wi-Fi and Bluetooth. Within the evaluation, we rely on Wi-Fi functionality for its extended range. Bluetooth-based dissemination in *BYPASS.KOM* was studied prototypically in [145].

DIRECT DISSEMINATION. Used for the direct ad hoc dissemination of local events, this mechanism proxy depends on the local transport proxy detailed before. It provides mechanisms for one hop and multi hop event dissemination and supports self healing as discussed in Section 4.3. During transitions from one

dissemination mechanism to another, no state information needs to be transferred for the currently implemented protocols.

CLIENT GATEWAY LOGIC. When acting as a gateway, a client needs to relay incoming events from the cloud to other nearby users. Relying on the stateless gateway approach published in [142] and previously discussed in Section 4.3.4, this proxy simply utilizes another instance of a proxy providing local dissemination mechanisms to forward messages to nearby clients.

GATEWAY DISSEMINATION. Functionally, this proxy is another instance of the direct dissemination proxy mentioned before. It operates on a separate port and is utilized solely by the client gateway logic to forward events to nearby clients. As such, it also relies on the local transport proxy to communicate via one of the available ad hoc network interfaces.

FILTER SCHEME. This proxy manages the client component implementation of the filter schemes introduced in Section 4.2.4. It provides the API for location-based publish/subscribe to the application. When executing a transition to another filter scheme, relevant state that needs to be transferred includes the current set of subscriptions and associated location requests. As filter schemes only communicate with their counterpart at the broker, they do not depend on any other mechanism proxy.

The broker component of `BYPASS.KOM` only utilizes two mechanism proxies: one for the utilized filter scheme and the other one for the utilized gateway selection algorithm.

FILTER SCHEME. This proxy manages the broker components of the location-based filter schemes discussed in Section 4.2.4. If total transitions are to be executed, it simply switches from one scheme to the other, taking care of state transfer and transformation. However, for partial transitions, multiple separate instances of proxies for filter scheme broker components are instantiated and managed.

GATEWAY SELECTION. This proxy acts as a wrapper for access to the generic framework of gateway selection algorithms provided in the `SIMONSTRATOR.KOM` based on the works in [142, 156]. It is used to execute transitions between different such algorithms during runtime. The interaction with the gateway selection framework is discussed in more detail in the following section.

The dependencies between mechanism proxies in `BYPASS.KOM` are illustrated in Figure 26 and further modeled in [189]. Solid black arrows indicate a direct utilization of a transition-enabled mechanism from within another mechanism proxy. This is, for example, the case for the gateway logic proxy, which directly utilizes the respective dissemination mechanism proxy. Also, both dissemination mechanisms (for local publications and gateways) utilize the local transport proxy to operate independent of the actual transmission technology used for ad hoc communication.

*Dependencies
between
mechanism
proxies*

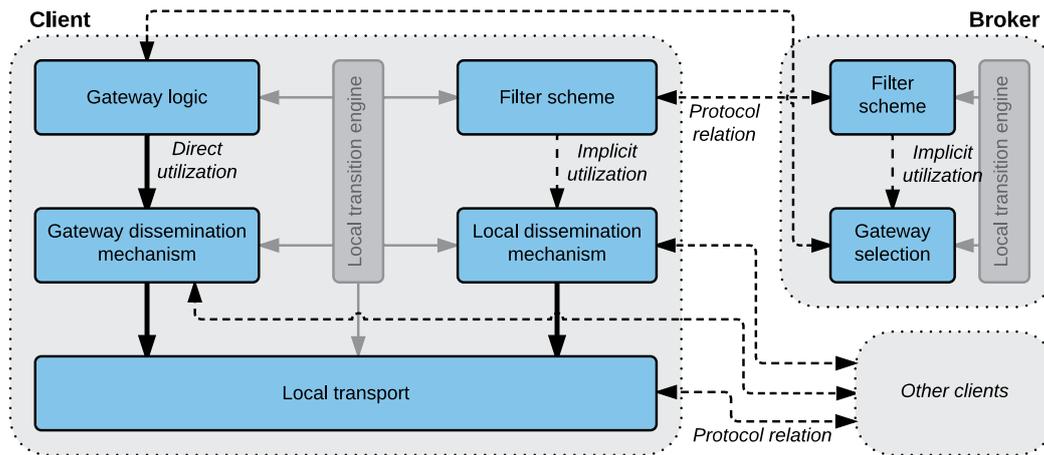


Figure 26: Mechanism proxies in BYPASS.KOM and their dependencies.

Besides direct utilizations, there exist two cases of an implicit utilization in BYPASS.KOM. In this case, the configuration of the publish/subscribe system determines whether there is a utilization or not. This is, for example, the case for the client's filter scheme mechanism: if direct local dissemination of events is configured, the respective events are passed to the local dissemination mechanism. This is, however, coordinated by the framework and not the mechanism proxy itself.

Implicit utilization

Given the nature of a distributed system, mechanism proxies operating on different devices have a protocol relation with each other. This is, for example, the case for the client and broker component of a transition-enabled filter scheme, given that the components communicate via a custom context update protocol. Additionally, the gateway selection (or, more specifically, the packing component as introduced in Section 4.3.4) has a protocol relation to the client gateway logic, given that relevant information needs to be encoded in the respective messages. Similarly, the transport and dissemination mechanisms utilized by different mobile clients exhibit a protocol relation, as only matching mechanisms are able to successfully exchange messages. In the following, we discuss the coordination of the mechanism proxies introduced in this section in the context of coexisting mechanisms.

Protocol relation

5.3.2 Coordination of Transitions for Coexisting Mechanisms

To combine location-based filtering and locality-aware dissemination of events we proposed the combined utilization of the respective transition-enabled mechanisms in Section 4.4. In the following, we discuss how the behavior defined in Section 4.4 is modeled and realized with *execution plans* and the *generic mechanism* M_{\emptyset} . Thereby, we show the applicability of the concepts and abstractions proposed in this chapter. Without loss of generality, we use concrete mechanisms for the discussion of the

execution plans to ease understanding. The mechanisms and parameters introduced in this section are also used during the evaluation presented in Section 6.4.

First, we discuss the realization of the scenarios *CX-ADHOC* and *CX-GATEWAY*. We then propose and model the adaptation of execution plans depending on the observed client density around an attraction point, showing the flexibility of our approach to adaptive event brokering for mobile social applications.

Ad Hoc Event Dissemination near Attraction Points

In addition to switching between filter schemes, we execute transitions to enable ad hoc dissemination of local events around attraction points. In the following, we present the resulting execution plans for the broker (EP_b) and the respective client (EP_u). We use *RADIAL* as representative parametric filter scheme for clients outside the area covered by attraction points. We assign a channel to each attraction point by using *ATTRACT* as channel-based filter scheme for clients near attraction points.

When client u approaches an attraction point, a partial transition $T^{\parallel}(u) : \text{RADIAL} \rightarrow \text{ATTRACT}$ is executed by the broker. Thereby, the state associated to the client is transferred between the respective filter schemes' broker components. Subsequently, the total transition $T : \text{RADIAL} \rightarrow \text{ATTRACT}$ needs to be executed by the client. Additionally, a local dissemination scheme is activated. In this example, we use *Gossip* with a message forwarding probability of $p = 0.4$. Consequently, we execute the transition $T : M_{\emptyset} \rightarrow \text{Gossip}$ at the client, followed by a self-transition $T^{\bullet} : f(\text{Gossip})$, with $f(\text{Gossip}) = p \leftarrow 0.4$ setting the message forwarding probability. The impact of p on the performance of *Gossip* is further evaluated in A.3. The resulting execution plans for the broker and the client are as follows:³

$$\begin{aligned} EP_b &= \left[T^{\parallel}(u) : \text{RADIAL} \rightarrow \text{ATTRACT} \right] \\ EP_u &= \left[T : \text{RADIAL} \rightarrow \text{ATTRACT}, T : M_{\emptyset} \rightarrow \text{Gossip}, T^{\bullet} : f(\text{Gossip}) = p \leftarrow 0.4 \right] \end{aligned}$$

As soon as the client detects locally that it is about to leave the area covered by the assigned channel within *ATTRACT*, it sends a context update message to *ATTRACT* to request a new assignment. This is reported to the coordinator component, which initiates the respective execution plans to switch the client back to the *RADIAL* scheme. Additionally, the local dissemination mechanism is turned off:

$$\begin{aligned} EP_b &= \left[T^{\parallel}(u) : \text{ATTRACT} \rightarrow \text{RADIAL} \right] \\ EP_u &= \left[T : \text{ATTRACT} \rightarrow \text{RADIAL}, T : \text{Gossip} \rightarrow M_{\emptyset} \right] \end{aligned}$$

Gateway-based Event Dissemination near Attraction Points

Given that *ATTRACT* is a channel-based filter scheme, events targeted at a location within a channel in *ATTRACT* need to be distributed to all clients within vicinity of

³ In addition to activating an ad hoc dissemination protocol, we need to activate the respective communication interface at the mobile device, as also discussed for our Android prototype in [145]. We omit these transitions for brevity.

the attraction point. Instead of sending the event to each client individually, gateways combined with local dissemination can be utilized to offload the cellular connection, as proposed in the *CX-GATEWAY* scenario. However, the gateway selection algorithms discussed in Section 4.3.4 require knowledge about a client's current position at the broker. In *ATTRACT*, locations of clients within a channel are usually not updated, as the clients themselves detect whether they are about to leave the area covered by a channel. Therefore, we include an additional self-transition when switching the filter scheme: $T^\bullet : f(\text{ATTRACT}) = \text{enable_updates} \leftarrow \text{true}$. The parameter `enable_updates` instructs the client component of *ATTRACT* to continue reporting location updates. We utilize the context update protocol of the filter scheme instead of a dedicated monitoring mechanism in our prototype and during evaluations, as the overhead in terms of message transmissions is negligible.

This configuration of *BYPASS.KOM* still disseminates local events via the *GOSSIP* mechanism as discussed in the previous section. To additionally enable dissemination of events from gateways to other clients nearby, we utilize a second instance of a transition-enabled dissemination mechanism. This is achieved by instantiating a second mechanism proxy that listens on a different port for incoming messages, as discussed in Section 5.3 for the prototype realization of *BYPASS.KOM*. As proposed in [142], gateways utilize the one hop dissemination mechanism *BROADCAST*. The dissemination mechanism is activated when a client enters a channel corresponding to an attraction point. This allows the client to listen to incoming messages sent by gateways nearby or to disseminate an event if selected as a gateway. Once the client leaves the area covered by the respective channel, it is instructed to deactivate the local dissemination mechanism as described earlier.

Given the fact that a client now uses two transition-enabled dissemination mechanisms in parallel, an execution plan needs to contain a reference as to which instance is affected by the transition. Therefore, we assign a name to each instance of a transition-enabled mechanism which is then used within the execution plan. This name corresponds to the name assigned to the mechanism proxy on creation, as discussed in Section 5.2.2. To apply a transition to a specific instance of a transition-enabled mechanism, we include the name of the instance. In the following, *gw* refers to the dissemination mechanism used by gateways and *lc* refers to the mechanism used for the direct dissemination of local events. Consequently, the following execution plan is sent to clients entering the channel of an attraction point:

$$EP_u = \left[T : \text{RADIAL} \rightarrow \text{ATTRACT}, T^\bullet : f(\text{ATTRACT}) = \text{enable_updates} \leftarrow \text{true}, \right. \\ \left. T_{lc} : M_\emptyset \rightarrow \text{GOSSIP}, T_{lc}^\bullet : f(\text{GOSSIP}) = p \leftarrow 0.4, T_{gw} : M_\emptyset \rightarrow \text{BROADCAST} \right]$$

When leaving the area covered by a channel, both local dissemination mechanisms are deactivated and the filter scheme is switched back to *RADIAL*:

$$EP_u = \left[T : \text{ATTRACT} \rightarrow \text{RADIAL}, T_{lc} : \text{GOSSIP} \rightarrow M_\emptyset, T_{gw} : \text{BROADCAST} \rightarrow M_\emptyset \right]$$

The execution plan at the broker does not differ from the previous case, as the broker only executes a partial transition of the filter scheme associated to the respective client. However, in contrast to the previous configuration, the broker needs

to utilize the gateway selection algorithm when disseminating events to clients subscribed in the ATTRACT filter scheme. Therefore, the `get_gateways` method discussed in Section 4.3.4 is used to select gateways and associated clients out of the set of subscribers. The gateway selection requires a parameter k denoting the (expected) number of gateways, as previously discussed. We select k based on the number of clients subscribed to the respective channel within ATTRACT and depending on whether a one-hop or a multi-hop dissemination mechanism is used. This is discussed in detail during the evaluation presented in Section 6.4.2.

The number of clients and their density varies depending on the popularity and size of an attraction point and may change over time. In the following section, we therefore utilize transitions to continuously adapt the choice of mechanisms used nearby an attraction point based on the observed client density.

Rule-based Adaptation of Utilized Mechanisms

In the previous configurations, transitions are only executed when clients enter or leave the area covered by a channel in ATTRACT. Once activated, the mechanisms utilized in the vicinity of an attraction point are not adapted to changing conditions during runtime. In this section, we discuss the execution of transitions to adapt the combination of mechanisms used near an attraction point based on the currently observed client density.

As previously discussed in Section 5.2.4, we focus on the execution of transitions rather than the reasoning on when to execute transitions. Therefore, we follow a simple rule-based approach for the execution of transitions, as discussed in the following. We select the set of mechanisms based on an assumed client density ρ . Whenever a client approaches an attraction point and subscribes to the respective channel in ATTRACT, the value of ρ is updated. The previous value of ρ is stored as ρ_{prev} .

Depending on ρ and ρ_{prev} , an execution plan for the adaptation of the current set of dissemination mechanisms is prepared by the broker. We refer to this execution plan as EP_a , with $\text{EP}_a = [T_{\text{lc}}, T_{\text{gw}}]$. In contrast to the previously discussed execution plans, the definitions of the transitions T_{lc} and T_{gw} now depend on ρ . We define T_{lc} such that PLAN-B is utilized for direct local dissemination if ρ falls below a threshold δ , and GOSSIP otherwise:⁴

$$T_{\text{lc}} : \begin{cases} M_{\emptyset} \rightarrow \text{PLAN-B} & \text{if } \rho \leq \delta \wedge \rho_{\text{prev}} > \delta \\ M_{\emptyset} \rightarrow \text{GOSSIP} & \text{if } \rho > \delta \wedge \rho_{\text{prev}} \leq \delta \\ \emptyset & \text{else.} \end{cases}$$

Note that instead of explicitly specifying the source mechanism, we use M_{\emptyset} as generic source in both cases. Thereby, the transition can be executed regardless of the client's currently running mechanism. As discussed in Section 5.2.2, the transition

⁴ When switching to GOSSIP, the self-transition $T_{\text{lc}}^{\bullet} : f(\text{GOSSIP}) = p \leftarrow 0.4$ might need to be added to the execution plan. Otherwise, the new mechanism instance of GOSSIP will use its default probability.

engine automatically determines the most suitable transition realization depending on the transition specification and the locally utilized mechanism.

Analogous to the definition of T_{lc} , we adapt the dissemination mechanism used by gateways based on ρ :

$$T_{gw} : \begin{cases} M_{\emptyset} \rightarrow \text{UNICAST} & \text{if } \rho \leq \delta \wedge \rho_{prev} > \delta \\ M_{\emptyset} \rightarrow \text{BROADCAST} & \text{if } \rho > \delta \wedge \rho_{prev} \leq \delta \\ \emptyset & \text{else.} \end{cases}$$

We use UNICAST to disseminate events from gateways to clients for lower densities. As soon as the density exceeds ρ , BROADCAST is used instead. This is motivated by our evaluation results published in [142]. A unicast-based transmission benefits from higher bandwidth due to the modulation scheme chosen as a consequence of the Request to Send (RTS) and Clear to Send (CTS) “negotiation”. Broadcasts rely on a more robust modulation scheme, thereby limiting the bandwidth of transmissions. However, as gateways need to send unicasts one after the other, the effective bandwidth decreases for increasing client density, as shown in [142].

The resulting execution plan EP_{α} is discarded, if it contains only empty transitions ($T_{lc} = \emptyset \wedge T_{gw} = \emptyset$). Otherwise, it needs to be executed by all subscribers U in the channel. As an alternative to notifying *all* subscribers using the transition coordinator, we also evaluate a strategy that relies on the self-healing capabilities of local dissemination mechanisms discussed in Section 4.3.3. Thereby, the execution plan only needs to be distributed to a single client, reducing the coordination overhead.

We evaluate the behavior resulting from coexisting transition-enabled mechanisms in Section 6.4. Our evaluation shows that the proposed execution plans enable the system to adapt to application-specific workload characteristics and client mobility.

The SIMONSTRATOR.KOM platform constitutes an important contribution towards generalizing the design of transition-enabled systems. Building on a rich foundation of domain-specific studies of transitions in video streaming [140, 161, 191], monitoring [153, 157], and publish/subscribe [142, 144, 147, 149], the abstractions provided by the platform support the design and evaluation of transition-enabled communication systems. Additionally, the clear separation of individual contributions into components fosters the re-use of existing mechanisms and benefits contributions in early stages of a project. The platform and its runtime environments form the basis for a growing number of demonstrators [143, 145, 150, 154, 172, 194], including demonstrations of dissemination mechanism transitions [145] and filter scheme transitions [143] in BYPASS.KOM.

EVALUATION OF MECHANISM TRANSITIONS

BASED on our prototype of `BYPASS.KOM` within the `SIMONSTRATOR.KOM` platform, we conduct an extensive evaluation of mechanism transitions in different aspects of publish/subscribe systems. Furthermore, we highlight the impact of key design decisions such as state transfer mechanisms and self-healing properties on the performance of the system during transitions. We detail our evaluation goals and methodology in the following, based on which we discuss our evaluation setup in Section 6.1. The setup includes the map-based mobility models utilized to capture real-world behavior of mobile clients as discussed in Section 6.1.1 and the corresponding application workload models presented in Section 6.1.2.

The evaluation addresses the mechanisms integrated into `BYPASS.KOM` and their transitions as presented in Chapter 4. Consequently, we evaluate the impact of transitions for location-based filtering in Section 6.2, followed by the evaluation of transition-enabled mechanisms for locality-aware event dissemination in Section 6.3. The combination of location-based filtering and locality-aware dissemination and the resulting coexistence of transition-enabled mechanisms is evaluated in Section 6.4.

The goal of this evaluation is to characterize mechanism transitions as a means to adapt a communication system within the application domain of location-based publish/subscribe. This includes understanding the impact of a transition execution on the system's performance during runtime. Given our research goal, we evaluate how transitions contribute to combining and composing mechanisms for location-based filtering and locality-aware dissemination in a highly adaptive fashion. In addition to the evaluation of concrete transitions between filter schemes and dissemination mechanisms as proposed in Chapter 4, we also assess the applicability of our generic design concepts for transition-enabled systems proposed in Chapter 5.

Concrete transitions and generic concepts

To characterize and isolate the impact of a transition execution, we need to compare the transition-enabled system against static mechanisms in a reproducible setup. At the same time, the setup needs to reflect the dynamics and heterogeneity of mobile social applications, as previously discussed. Achieving a reproducible setup in real-world measurements is hard to achieve, especially at the scale required to assess the dynamics of mobile social applications. Therefore, we rely on event-based simulations of our proposed system in the `SIMONSTRATOR.KOM` platform, utilizing models for client mobility derived from earlier studies [7, 145, 186]. The evaluation setup and the utilized models are further discussed in the following section.

6.1 EVALUATION SETUP

We evaluate our proposed mechanisms and their transitions using the prototype of BYPASS.KOM implemented in the SIMONSTRATOR.KOM platform as described in Section 5.3. We use an updated version [155] of the event-based simulator Peerfact-Sim.KOM [177] as underlying simulation engine. The updated version contains additional mobility models specifically targeted at our application scenario, as detailed in the following section. To evaluate the publish/subscribe system under realistic workloads, we propose workload models that capture key communication characteristics of our scenario. The workload models are described in Section 6.1.2. The network model utilized for our evaluation is detailed in Section 6.1.3. Finally, the evaluation metrics used to capture the behavior of transition-enabled mechanisms are introduced in Section 6.1.4, including a brief introduction of the measurement methodology and the plot types used in this chapter.

6.1.1 Mobility Models

Given the scenario of mobile social applications, human mobility needs to be accurately modeled in our evaluation [37, 63]. To this end, we extended our earlier work on modeling human mobility [155, 177, 186]. By integrating the corresponding mobility models into the SIMONSTRATOR.KOM platform, we evaluate our proposed transition-enabled publish/subscribe system under realistic client mobility, employing real-world map data. For comparison, we also conduct evaluations with the generic Random Waypoint Model [81]. This supports comparing our results to other state of the art approaches that still rely on simple mobility models [98]. Additionally, it provides insights on how an accurate mobility model exposes important characteristics of a communication system that would be left unnoticed in a generic scenario. The mobility models discussed in this section are summarized in Table 2. For each mobility model, we describe the *local movement* strategy (i. e., the model used to approach a target location) and the strategy used to select a *target location* in the following.

MOBILITY MODEL	LOCAL MOVEMENT	TARGET LOCATION
DA	Map-based, Darmstadt	Attraction points Darmstadt
MA	Map-based, Mannheim	Attraction points Mannheim
DA-RND	Map-based, Darmstadt	Random location
MA-RND	Map-based, Mannheim	Random location
RWP	Linear	Random location

Table 2: Mobility models and their characteristics.

Map-based Mobility

Based on our earlier work [155, 177], we rely on real-world map data to model client mobility. We utilize OpenStreetMap (OSM) as map data provider, and the GraphHopper open-source library¹ for navigation. Relying on the routing capabilities of GraphHopper, clients follow pedestrian walkways to reach their target destination. The target destination is either selected randomly or among the set of attraction points defined by the application. As soon as a client reaches its target destination, the client pauses for a random time t_p , randomly chosen from the uniformly distributed interval between $t_{p,\min}$ and $t_{p,\max}$, before determining the next target location.

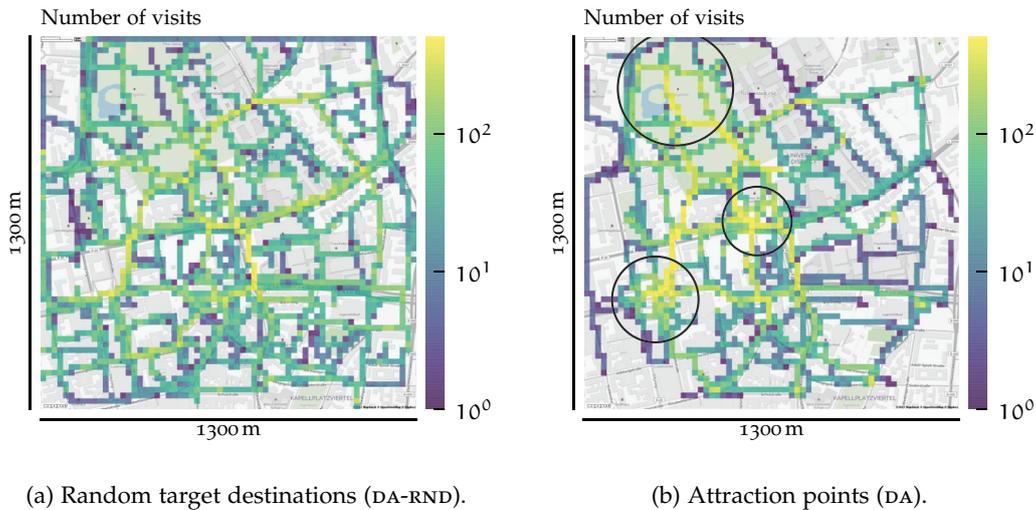


Figure 27: Client mobility for the city center of Darmstadt.

The resulting mobility characteristics are shown in Figure 27a. The plots show the accumulated visits of clients over a time period of 30 minutes in the simulated area of $1,300 \times 1,300$ meters in the city center of Darmstadt, Germany. Visits are aggregated cells of 20×20 meters. Client mobility is limited to pedestrian walkways, as shown in Figure 27a. For random target destinations, the number of visits is spread more evenly across the simulated area, including visits to more remote areas. If attraction points are chosen instead, the number of visits in vicinity of attraction points increases, while remote areas of the map are not visited at all, as shown in Figure 27b.

To capture the essence of more grid-like mobility models such as the Manhattan model proposed in [5], we conduct additional evaluations based on the map of the city center of Mannheim, Germany. The resulting mobility characteristics are shown in Figure 28a and Figure 28b, clearly highlighting the grid-like nature of Mannheim's city center. In comparison to Darmstadt, visits are more equally distributed across the simulated area. This is due to the fact that within the grid-like structure of Mannheim, the GraphHopper navigation library tends to find multiple different routes with similar length for a target location.

¹ www.graphhopper.com/open-source/ [Accessed March 8th, 2017].

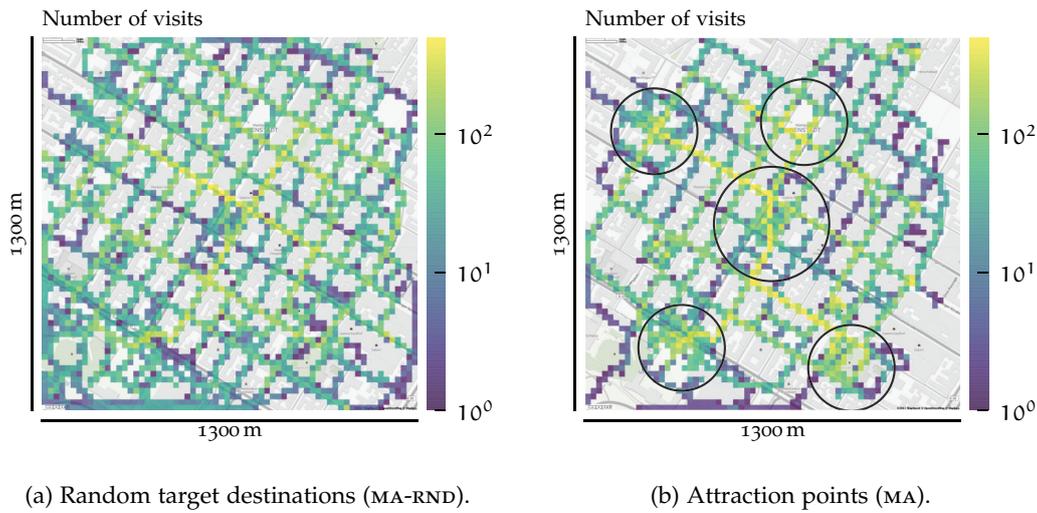


Figure 28: Client mobility for the city center of Mannheim.

For both cities, attraction points are placed at public parks according to OSM data. All attraction points have an equal probability of being selected as a client’s target location, referred to as the attraction point’s *weight*. To prevent clients from targeting the exact same locations, we add a random offset from the center location of the attraction point. This offset is modeled via two Gaussian distributions with $\mathcal{N}_x(x_0, \sigma^2)$ and $\mathcal{N}_y(y_0, \sigma^2)$ that are used to draw the x and y coordinate around the attraction point (x_0, y_0) . To ensure that more than 99% of the resulting target locations are within a radius r around the current attraction point, we set $\sigma = r/3$. For the attraction points generated from OSM data, we assume a default radius of $r = 25$ meters to ensure that clients approach slightly different target coordinates.

To capture the real-world behavior of clients in augmented reality applications accurately, we manually assign higher probabilities and larger radii to a subset of the attraction points. Consequently, a greater number of clients visit the vicinity of these attraction points. To this end, we utilize data on in-game attraction points that is openly available for the augmented reality games Ingress and Pokémon Go. For Ingress, the location of *Portals* and their significance is available on Ingress Intel.² The location of in-game *Gyms* in Pokémon Go is gathered and provided by PokémonGoMap.³ Both data sources motivate our choice of manually weighted attraction points for Darmstadt and Mannheim. Given the characteristics of the respective applications, these attraction points coincide with pedestrian zones and public parks. These manually altered attraction points are shown in Figure 27b and Figure 28b as black circles indicating their radius.

² www.ingress.com/intel, Google account required [Accessed March 8th, 2017].

³ www.pokemongomap.info [Accessed March 8th, 2017].

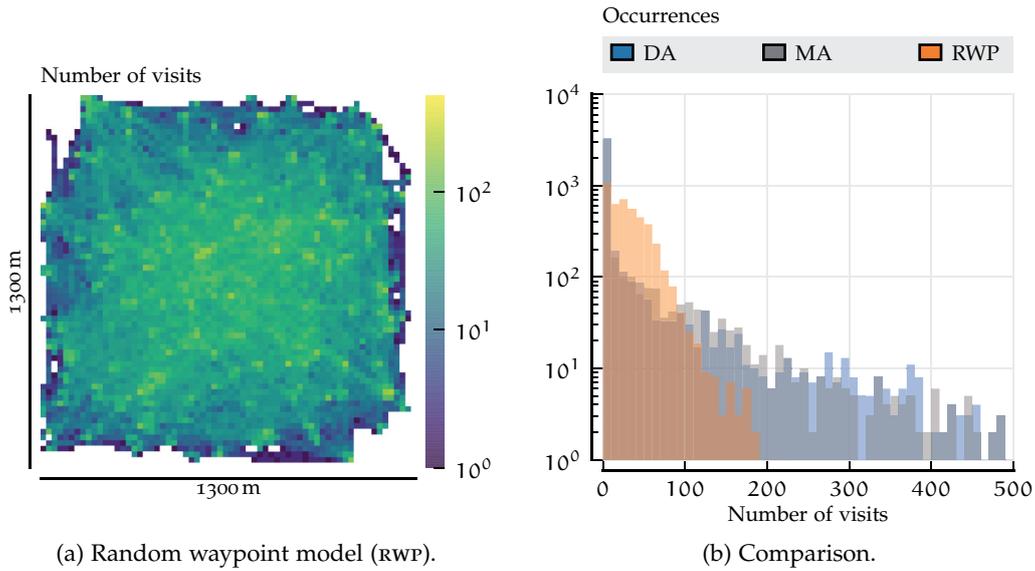


Figure 29: Random waypoint mobility compared to map-based mobility.

Random Waypoint Mobility as Baseline

To assess the impact of the mobility models in comparison to a baseline frequently used in related works, we include the Random Waypoint Model [81] in our evaluation. Here, clients select a random target location and approach that location following a straight line. After reaching their target location, they pause for a time t_p before continuing to a new randomly chosen target location. As for the map-based models, t_p is uniformly distributed between $t_{p,\min}$ and $t_{p,\max}$.

The resulting mobility characteristics are shown in Figure 29a. The visits are distributed more evenly across the simulated area, with areas in the center being visited more often than the outskirts. Target locations with longer pause times are also clearly visible. Figure 29b shows a histogram of the number of times a cell is visited for the map-based models of Darmstadt (DA) and Mannheim (MA) and the Random Waypoint Model (RWP). In comparison to the map-based models, the visits are more evenly distributed and there are no locations with more than 200 visits. However, the nature of human mobility (especially in an urban scenario) is not accurately captured with the Random Waypoint Model, as also pointed out in more recent research on human mobility models [77, 185].

All models are configured with the parameters listed in Table 3. This includes the size of the simulated area, the movement speed that is uniformly distributed between v_{\min} and v_{\max} for pedestrian mobility [90], and the time a client spends at the target location as previously explained. We select the size of the simulated area and the number of nodes according to the setup discussed in [175], motivated by a study by Kurkowski et al. [99] regarding the standardized evaluation of MANET routing protocols. However, as the respective studies are limited to random waypoint

PARAMETER	DEFAULT VALUE
World size	1,300 m × 1,300 m
Number of mobile clients	297
Movement speed [v_{\min} , v_{\max}]	[1 m/s, 2 m/s]
Pause times [$t_{p,\min}$, $t_{p,\max}$]	[1 s, 60 s]

Table 3: Mobility parameters used in all models.

mobility, the respective characteristics in terms of node density and connectivity do not necessarily hold for our more complex mobility models. Therefore, we use the average number of subscribers resulting from our mobility models in combination with the workload models to characterize our simulation setup, as discussed in the following section.

6.1.2 Workload Models

Depending on the application at hand, interest in content can be closely tied to users' locations and their surrounding attraction points. In any case, users are interested in all events that are published to a location within a given distance from their current location. Within the publish/subscribe system, this interest is denoted via the `subscribe` method defined in Section 4.2.1, with the maximum distance being specified as the subscription radius. Each client periodically publishes an event to a target location. To assess the implications of different locality patterns, we define the following distinct workload patterns for a location-based mobile social application.

- AR. The AR workload mimics communication characteristics of a mobile augmented reality game, with events being published to the user's current location. Consequently, only users in close proximity are interested in the event, leading to a high locality of content and interest. In this model, all events are published through the `publish_local` method as defined in Section 4.2.1.
- LBS. The LBS workload models characteristics of a more generic location-based service. Clients publish events to specific attraction points rather than to their own location. Consequently, clients within proximity of the respective attraction point are interested in the event. This model does not exhibit strong locality in content and interest. However, as events are being published to the locations of attraction points, the average number of interested subscribers is high. This is due to the fact that clients tend to gather around attraction points, as discussed in the previous section.
- RND. As a generic baseline, we also consider a completely random choice of target locations. Consequently, some events might not have any subscribers at all, depending on the distribution of clients in the simulated area. For RND, all events are published through the `publish` method rather than through `publish_local`.

PARAMETER	DEFAULT VALUE
Location update interval	5 s
Publication frequency per client	1/s
Application payload (added to events)	128 byte
Radius of the AoI	125 m

Table 4: Additional workload parameters.

As discussed in [155], load on the communication system is directly determined by the mobility of clients in conjunction with the behavior of the selected workload model. Static configuration parameters used in all workload models are summarized in Table 4 and discussed in the following.

The location update interval describes the time between two consecutive location updates of a client’s location sensor. This interval determines the maximum accuracy of location-based filter schemes, as discussed in Section A.1. The publication frequency is fixed to one publication per second per client. As the goal of this evaluation is the characterization of mechanism transitions and not a performance evaluation of individual mechanisms, we do not further vary the publication frequency. As individual mechanisms are already evaluated by their respective authors, we instead refer the interested reader to the respective publications.

The same holds true for the application payload added to outgoing events. Varying the payload has an influence on the performance of local dissemination mechanisms. When discussing the offloading effects resulting from the hybrid utilization of gateway selection and dissemination mechanisms, we compare the relative utilization of the respective mechanisms. Therefore, the concrete payload size does not have an impact on our reported results. Given that the frequency of events is fixed for each client, the load on the system increases significantly with increasing client density, as events have to be distributed to a potentially larger number of interested subscribers. With the workload models presented in this section, we assess the impact of these effects on a number of mechanisms within BYPASS.KOM.

As motivated previously, we use the average number of subscribers for individual publications to characterize our simulation setup. We show the respective spatial distribution for the DA and MA mobility model in Figure 30, using the AR workload. The average number of subscribers varies significantly depending on the target location of a publication. Areas associated with attraction points (as indicated by the black circles) exhibit a higher client density and, consequently, higher number of interested subscribers. For both workloads, up to 40 clients within proximity are interested in a given publication, corresponding to approximately 13 % of the total client population. Rather than focusing on the absolute number of subscribers, we are more interested in the heterogeneity of this figure, as it motivates application-specific adaptation of the utilized mechanisms.

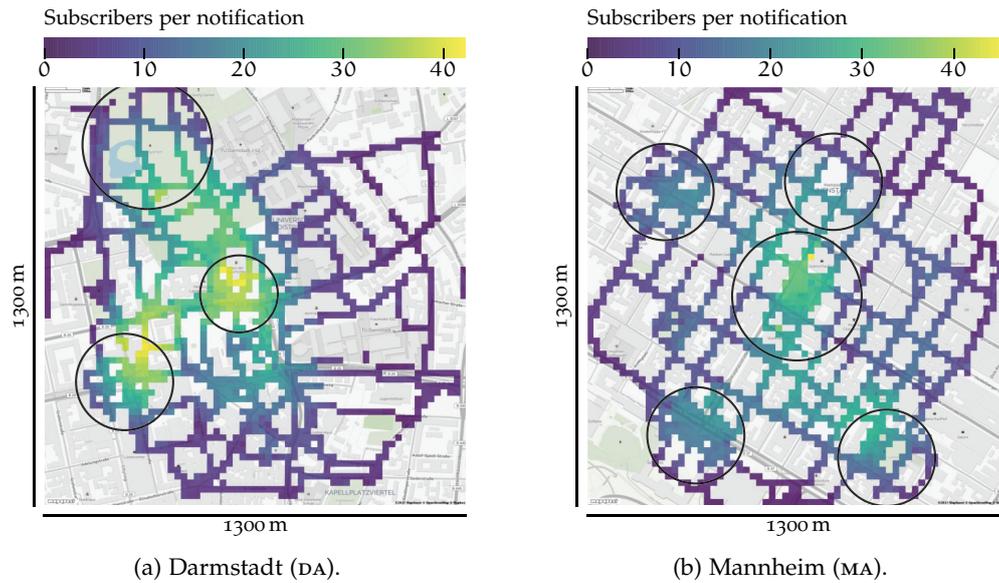


Figure 30: Average number of subscribers based on the publication location.

6.1.3 Network Model

To accurately model the scenario of mobile social applications, our simulation setup needs to provide two types of network models: *i*) a model for direct ad hoc communication between mobile clients and *ii*) a model for the communication with a cloud-based back end via a cellular connection.

*802.11 Wi-Fi
model for ad
hoc communi-
cation*

For the direct ad hoc communication, we rely on the 802.11 Wi-Fi Medium Access Control (MAC) model with distributed coordination function as provided by the NS3 network simulator [158]. This model has been integrated into the PEERFACT-SIM.KOM runtime within the SIMONSTRATOR.KOM platform as published in [147, 148]. The model relies on a simulation of the noise floor caused by concurrent transmissions to determine if packets can be transmitted. For unicast transmission, the respective RTS and CTS messages are exchanged before the actual data transmission occurs. For a more detailed description and analysis of the distributed coordination function of the 802.11 Wi-Fi MAC the interested reader is referred to [10]. The model relies on a log-distance path loss model configured with a path loss exponent of 3.8 for an urban environment as measured in the studies presented in [2, 111]. This configuration leads to an effective transmission range of around 88 m, while increasing the noise floor and, thus, the probability for collisions, for clients within up to 205 m range. As the chosen model takes the noise floor into account, it accurately captures the effect of increased client density and concurrent transmissions. Capturing this effect is crucial within the scope of our work, given that we utilize direct ad hoc communication at or around application-specific attraction points.

While an accurate model for the direct ad hoc connections between mobile clients is necessary to evaluate our work, we do not require the same level of accuracy

for our model of the cellular network. Instead, we rely on a measurement-based approach for our model of the cellular network. A client's cellular connection to the cloud-based broker network is assumed to be reliable, which is a reasonable assumption in city centers as long as the mobile network is not overloaded. The model simply assigns an estimated bandwidth and latency to the respective connection based on real-world measurements presented and collected in [82, 83, 117, 118]. For technical details of the respective model, the interested reader is referred to our publication [155]. We assume that the resources of the cellular network are sufficient to cater for all mobile clients in our simulations. This is motivated by the fact that we are interested in the offloading effects that are achievable with our locality-aware event dissemination mechanisms rather than in the network behavior in overload situations. In our simulations, the cellular network has an average latency of 200 ms and offers reliable connectivity with a guaranteed bandwidth of 1 Mbps per client. We configured the bandwidth such that it does not impact the operation of our system (i. e., it ensures proper handling of the cellular traffic caused and consumed by mobile clients).

6.1.4 Evaluation Metrics

We utilize a number of metrics to capture the impact of mechanism transitions and the performance and cost characteristics of individual mechanisms. On the one hand, we capture performance and cost metrics specific to publish/subscribe systems, such as *recall* and *precision* of event delivery. On the other hand, we measure network load in terms of traffic for both, clients and the cloud entity, to assess the resource utilization of the system. Additionally, we capture mechanism-specific metrics to characterize the utilization of gateways where it is appropriate. Finally, to capture the effects of transitions on a system, we measure their execution time and location. In summary, the following metrics are gathered and reported in our evaluation:

RECALL. The recall is defined as the ratio of correctly notified subscribers to all subscribers for a given event. Within our scenario, the valid set of subscribers for a given event is defined by the circular AoI provided by the application through our API. It is therefore independent of the actual filter scheme and should be close to 1.0 for all utilized mechanisms in order to satisfy the application requirements.

PRECISION. The precision complements the recall in characterizing a publish/subscribe system. It is defined as the ratio of correctly notified subscribers to all clients that were notified, regardless of whether they are actually subscribed to the event.

DELIVERY DELAY. We measure the average time it took to forward an event from its publisher to all reached subscribers. The delivery delay strongly depends on how the network latency and bandwidth are configured, which is why the absolute delay is not as important as the relative increase or decrease for different dissemination mechanisms.

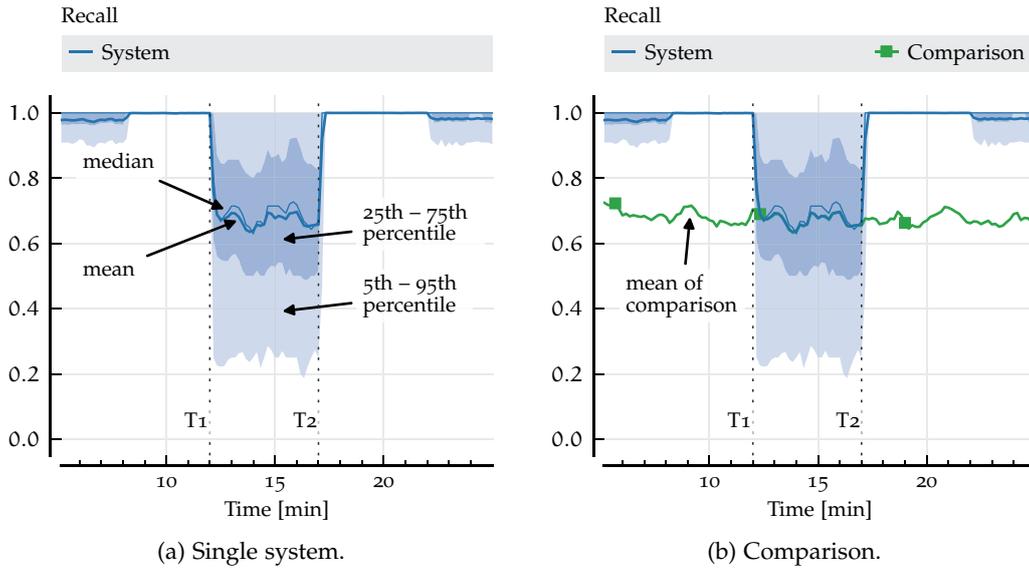


Figure 31: Sample plots showing measurements over time.

TRAFFIC. Given the challenge to efficiently utilize resources of mobile clients and the cellular network, we measure the traffic characteristics of our system. We distinguish between traffic over the cellular interface and traffic caused by direct local ad hoc communication. Additionally, we measure the traffic caused by a filter scheme’s context update protocol and the traffic resulting from the coordination of transitions.

SPREAD OF TRANSITIONS. To understand the behavior of transitions, we capture the time of their local execution on each affected client together with the currently utilized mechanism. We further keep track of a client’s current location when executing transitions, allowing us to characterize the spread of transitions within a scenario.

GATEWAY UTILIZATION. We introduced local gateways in order to relieve the cellular network in Section 4.3.4. To assess the fairness characteristics of the respective selection algorithms, we measure how often and for how long clients are utilized as gateways and how many other clients are served by them.

To assess the behavior of transitions during runtime, metrics are reported over time. Two sample plots are shown in Figure 31. In Figure 31a, the mean of the metric (in this case, the recall) is plotted over time. The mean recall, plotted as a solid line, is calculated over all events that occur within ten seconds. Additionally, the shaded areas report the percentiles: the darker shaded area includes all values between the 25th and 75th percentile, and the light area includes values between the 5th and 95th percentile. The median (50th percentile) is included as a thin solid line for reference.

While this format is an accurate representation of value distributions over time, it is not well suited to compare a larger number of systems against each other. There-

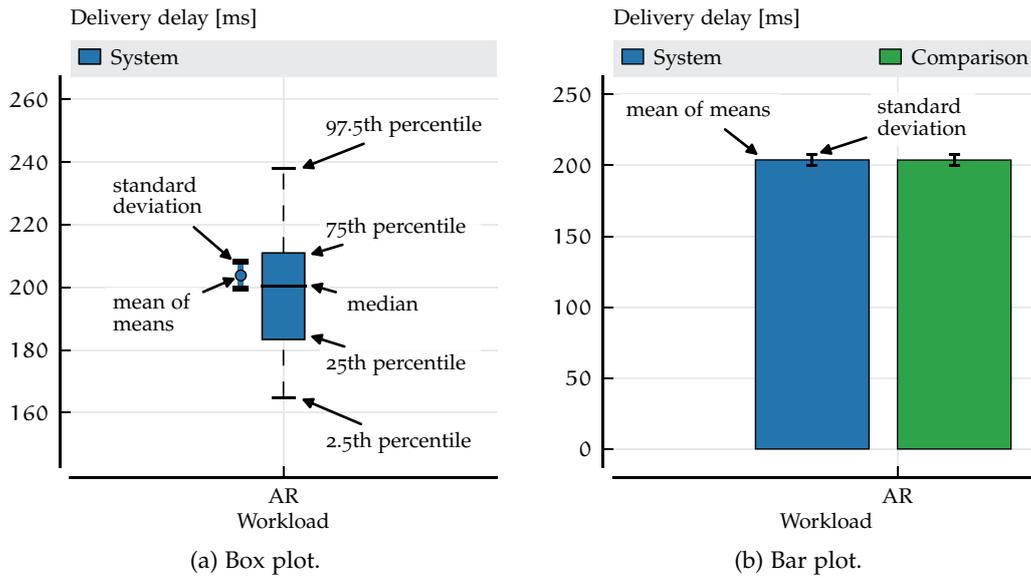


Figure 32: Sample plots for aggregated values

fore, systems included for comparison are often only plotted with their mean value, as shown in Figure 31b. Both plots further include dashed vertical lines that mark the execution time of a transition affecting all clients in the system.

Whenever aggregated results are reported, box plots or bar plots are utilized, as shown in Figure 32. The extent of the box indicates the 25th and 75th percentile, with the solid line indicating the median. Whiskers report the 2.5th and 97.5th percentile, meaning that 95% of the observed values lie within the whiskers. Boxes report the data gathered during one run of an experiment, corresponding to one random seed. To report the behavior of the metric for the remaining repetitions, we plot a circle indicating the grand mean (mean of the mean) over all runs. Additional whiskers around the circle indicate the standard deviation of the means for the individual runs of an experiment. Bar plots report the mean of the mean over multiple runs, with whiskers indicating the standard deviation of the individual mean values. Additional plot types are explained on their first occurrence in this chapter.

6.2 TRANSITIONS BETWEEN FILTER SCHEMES

In this section, we evaluate the impact of centrally coordinated total transitions between filter schemes during runtime of the publish/subscribe system (Section 6.2.1). We focus on the impact of our proposed state transfer mechanisms (Section 6.2.2) and the potential of self-transitions as generic representation of scheme-specific parameter adaptations (Section 6.2.3).

In addition to total transitions, we study partial transitions between filter schemes based on application-specific knowledge about attraction points. In Section 6.2.4, we compare the transition-enabled system to a self-adaptive filter scheme that realizes

the same filter behavior. This allows us to verify correct operation of our transition-enabled mechanisms and discuss the benefits and cost of mechanism transitions compared to self-adaptive mechanisms. An in-depth evaluation of the individual filter schemes realized within `BYPASS.KOM` and their parameter configurations is presented in A.1.

6.2.1 Centrally Coordinated Total Transitions

First, we evaluate total transitions between filter schemes as initially discussed in [144]. A total transition affects all clients, as defined in Section 4.2.3. We expect that the transition-enabled system utilizing a filter scheme *A* behaves like a static configuration of filter scheme *A*. After a transition to scheme *B*, we expect the overall system characteristics to reflect those of scheme *B*. We evaluate whether `BYPASS.KOM` meets these expectations and, thus, correctly executes transitions between filter schemes. The respective transitions are triggered at predefined times at the broker. The broker sends a corresponding execution plan to all currently subscribed clients. The transitions and the time of their execution are listed in Table 5.

#	TIME	EXECUTION PLAN
T1	8 min	[T : RADIAL → STE, T [•] : f(STE) = α ← 1.5]
T2	12 min	[T : STE → GRID, T [•] : f(GRID) = grid ← 5]
T3	17 min	[T : GRID → EGRID, T [•] : f(EGRID) = grid ← 10]
T4	22 min	[T : EGRID → RADIAL]

Table 5: Execution plans for the evaluation of total transitions.

Recall and precision for the transition-enabled system are shown in Figure 33 for the AR workload. For comparison, static configurations of the respective filter schemes without transitions are also included in the plots. The recall does not vary much between the different schemes, except for the grid-based scheme `GRID` that has an average recall of 0.7. After switching to `STE` (T1) or to `EGRID` (T3), the recall reaches 1.0, while for `RADIAL` approximately 25% of the clients experience a recall below 1.0, yet larger than 0.9. The precision of the transition-enabled system clearly reflects the characteristics of the individual filter schemes, as shown in Figure 33b. As expected, the system adapts its characteristics to the respective filter scheme right after a transition occurs.

In Figure 34, we show the traffic caused by the filter schemes' context update protocol. Traffic is shown over time to compare the performance of the transition-enabled scheme to the performance of the individual schemes after the execution of each total transition. Soon after each transition, the traffic caused by the context update protocol equals that of the currently active mechanism. However, we can observe sudden increases in traffic right after transitions T2 and T3 for the download direction and transitions T3 and T4 for the upload direction. This effect is caused by

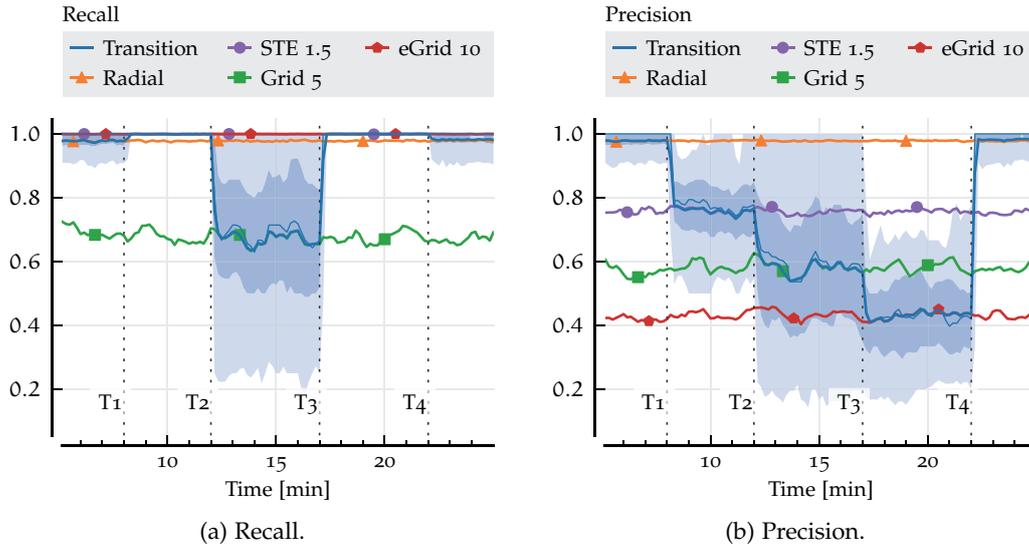


Figure 33: Recall and precision of the transition-enabled system over time.

the bootstrap phase of the respective transitions. After transition T2 from STE to GRID the broker reports the assigned channel to each client, causing the sharp increase in context protocol download traffic for the client. For the same reason download traffic increases right after transition T3, as in eGRID the broker also sends assigned channels to clients. As each client is assigned to more than one channel in the eGRID scheme, the initial download is slightly larger due to increased message sizes.

Considering the upload direction shown in Figure 34b, we observe an increase right after transition T3 and a slight increase right after transition T4. This is the consequence of missing state information at the broker, as discussed in detail in Section 6.2.2. The broker requires updated location information after T3 and T4 for channel assignment (T3) and general operation of RADIAL (T4). However, in both cases the previous filter scheme cannot provide the necessary information as neither GRID nor eGRID maintain updated client locations. Consequently, clients need to report their location if the assignment at the broker deviates too much from their current location, leading to the traffic increase right after T3. The effect is less visible for transition T4, as in RADIAL clients nevertheless need to send periodic location updates to the broker.

6.2.2 Impact of the State Transfer Mechanism

An important phase during the execution of a transition is the local state transfer between two mechanisms. As discussed in Section 4.2.3, state includes the subscriptions stored at the broker component of the respective filter scheme and any additional contextual information that has been gathered by the filter scheme. We compare the performance of the transition-enabled system with state transfer against

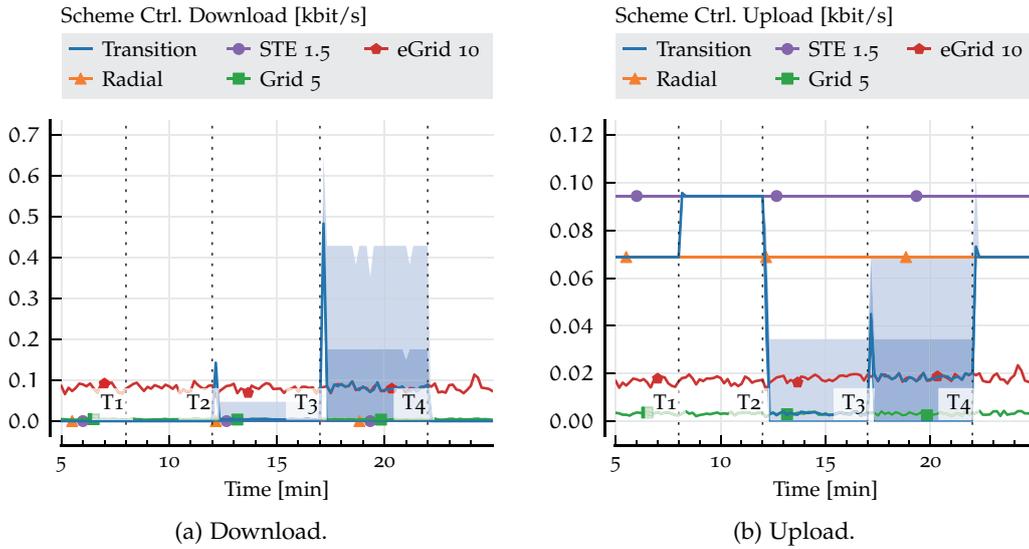


Figure 34: Traffic characteristics of the transition-enabled system.

transitions without state transfer for the AR and LBS workload models, assessing the difference between handling of local and generic location-based events during a transition. Without state transfer, clients need to re-subscribe at the broker and filter schemes need to be bootstrapped afterwards. Transitions are again triggered based on the elapsed time, with the execution plans given in Table 5.

Figure 35 shows the recall of both configurations over time right after each transition is triggered at the broker. Three cases can be distinguished and are discussed in the following: *i*) the recall remains constant due to state transfer (T1 and T2), *ii*) the recall drops to 0 for the AR workload even with state transfer in place (T3), or *iii*) there is a slight quality degradation during the transition (T4). In the first case, shown in Figure 35a and Figure 35b, the last known location of a client is information that is maintained by the original scheme's broker component *and* helps in bootstrapping the target scheme. After transition T1 from RADIAL to STE, the initial space time envelope is simply calculated with no additional movement vector, leading to a circular AoI. As soon as clients report their movement vector, the broker updates the shape of the envelope. This, in turn, leads to an increase in achieved recall, as expected for the STE scheme (refer to Section A.1 for an in-depth evaluation of the individual schemes' properties). During the transition T2 from STE to GRID, information about the last known location of a client that is available within the STE filter scheme can directly be utilized for the channel assignment. Consequently, there is no intermediate drop in recall if state information is utilized, as shown in Figure 35b.

After transition T3, the recall drops to 0 for a period of roughly 100 ms with state transfer, recovering afterwards. However, this is only the case for the AR workload publishing only local transitions. For generic location-based events, as created in the LBS workload, the state transfer mechanism is able to maintain the recall. This is due

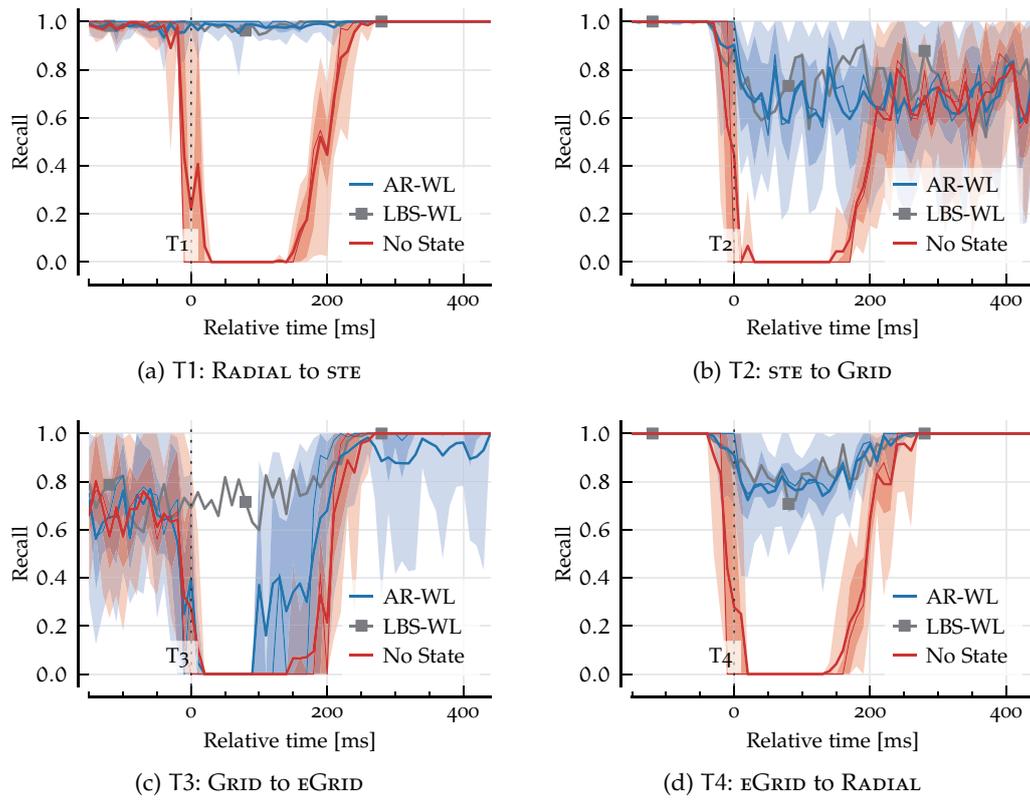


Figure 35: Impact of state transfer with AR and LBS workloads during transitions.

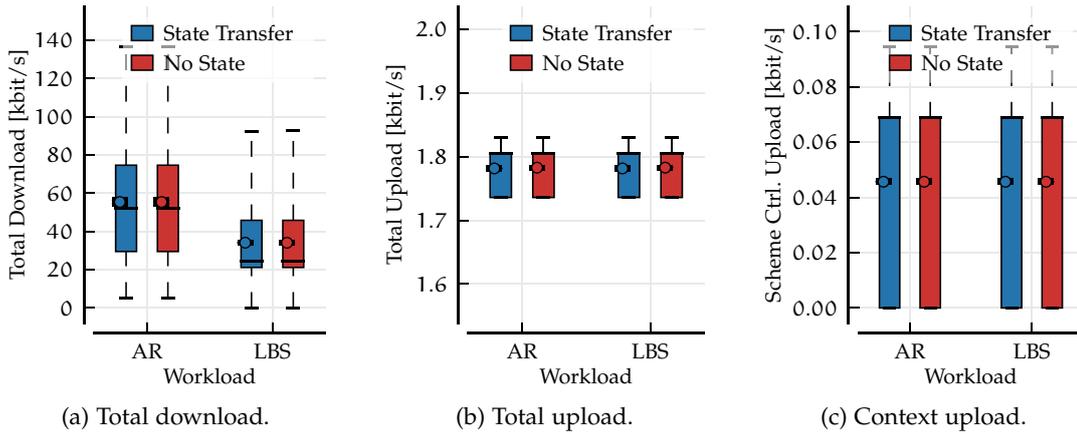


Figure 36: Client-side traffic with and without state transfer.

to the fact that the channel assignment changes from GRID to eGRID. Consequently, local events published to a specific channel in GRID can no longer be processed by the broker component of eGRID. For generic location-based events, assignment to channels is done by the broker. Given that the last known location of clients is transferred from GRID to eGRID, the broker can already use the updated channels of eGRID to notify relevant clients.

*Consequence
of outdated
state
information*

During transition T4, the slight quality degradation is caused by outdated state information being transferred to the RADIAL scheme. As eGRID only updates the last known location of a client as consequence of the client leaving its current cell, the location used as the center of the AoI in RADIAL is likely to be outdated. However, as soon as context update messages arrive at the broker, the recall reaches 1.0 again.

*No significant
overhead due
to state
transfer*

The state transfer mechanism does not affect the overall traffic compared to re-subscriptions, as shown in Figure 36a and Figure 36b. This is due to the fact that the traffic caused by context update messages and re-subscriptions is negligible compared to the traffic caused by client events and application payload within our scenario. The context update messages sent by clients account for only 2.5% of the total upload, as shown in Figure 36c. This value depends on the application payload included within events and on the frequency at which events are published. Given the asymmetry in up- and download traffic in our scenario, the download of context update messages accounts for even less overhead compared to the upload.

6.2.3 Self-Transitions for Filter Scheme Adaptation

In addition to global transitions, we propose self-transitions as a generic design concept to adapt a mechanism's configuration. Within this section, we evaluate self-transitions of the grid-based filter scheme eGRID within the AR and LBS workload. Therefore, the execution plans as shown in Table 6 are triggered at the specified times. Initially, clients use the eGRID filter scheme with a grid factor of 2, leading

#	TIME	EXECUTION PLAN
T1	8 min	$[T^\bullet : f(\text{EGRID}) = \text{grid} \leftarrow 5]$
T2	12 min	$[T^\bullet : f(\text{EGRID}) = \text{grid} \leftarrow 10]$
T3	17 min	$[T^\bullet : f(\text{EGRID}) = \text{grid} \leftarrow 20]$
T4	22 min	$[T : \text{EGRID} \rightarrow \text{RADIAL}]$

Table 6: Execution plans for the evaluation of self-transitions.

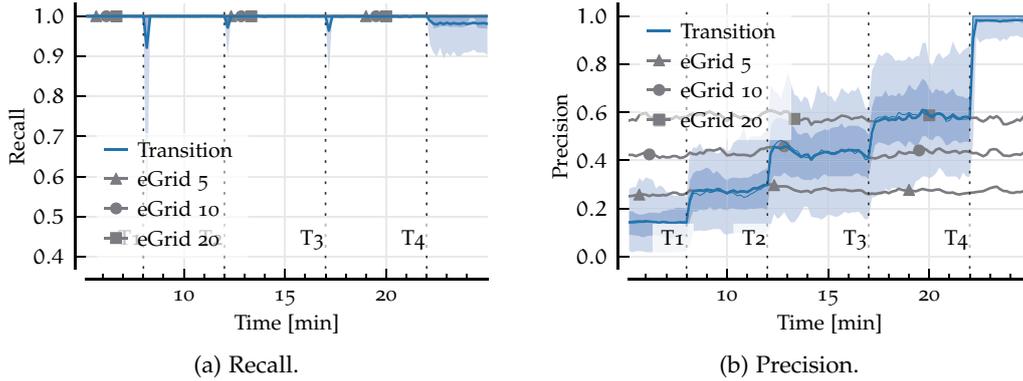


Figure 37: Recall and precision with self-transitions compared to static configurations.

to four cells. We discuss how updating the channel assignment function can lead to performance degradation during a transition for locally relevant events.

The resulting macroscopic system behavior under the AR workload compared to a static configuration of `EGRID` with the given grid factors is shown in Figure 37. The transition-enabled system is able to adapt its characteristics to those of the individual configurations, as previously shown for total transitions. As expected for `EGRID`, the recall remains 1.0, until we execute a global transition to the `RADIAL` filter scheme (T4), as shown in Figure 37a. However, there are slight quality degradations right after transitions are executed, clearly visible for T1 and T3. The precision follows the exact characteristics of static configurations as shown in Figure 37b, leading to a stepwise increase for smaller cells caused by higher values of the grid factor.

To identify the reasons for the quality degradation right after each transition, we take a closer look at T1 and T3 under both workloads, AR and LBS. From Figure 38 it becomes clearly visible that the performance is only degraded for the AR workload, whereas the performance for the LBS workload is not affected. This is due to the fact that events published to the client’s own location (as issued within the AR workload) are published to the channel where the client is currently located. This ensures that all subscribers registered for that channel receive the event, as discussed in Section 4.2.2. However, as a consequence of the self-transition, the channel assignment is updated. This, in turn, leads to new channel identifiers being issued by the broker. Until clients receive the updated channel identifiers via the context update

*Missing
channels for
local events*

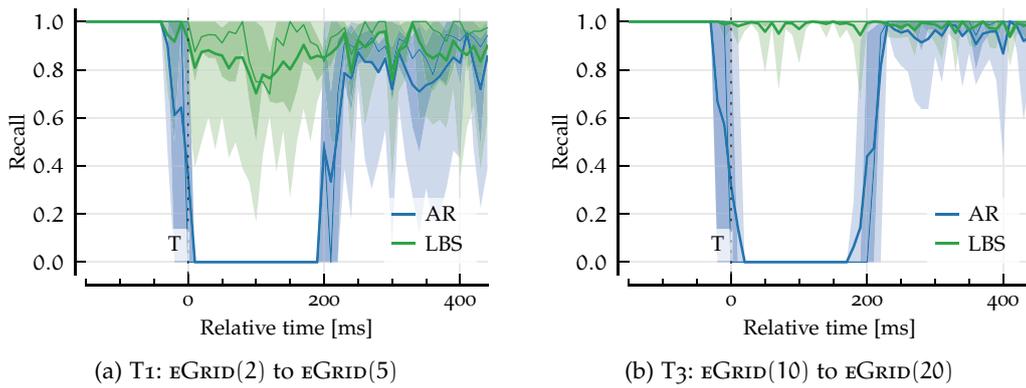


Figure 38: Recall of eGRID during self-transitions T1 and T3.

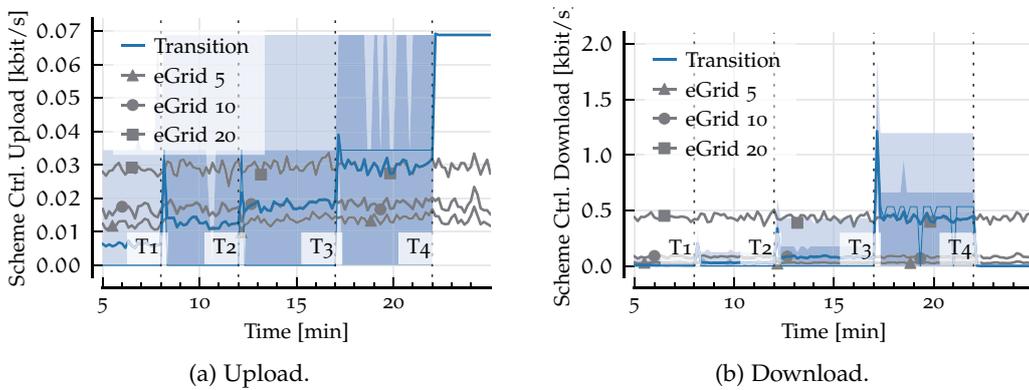


Figure 39: Client-side control traffic caused by self-transitions.

protocol, incoming events using the outdated channels are simply discarded. The duration of this effect is determined by the latency of the cellular connection, which is configured to around 200 ms, as discussed in Section 6.1.3.

For non-local events as issued by the LBS workload, the assignment to a target channel is not performed by the client, but instead by the broker. Therefore, incoming events can directly be processed with the updated assignment function and forwarded to the correct subscribers. As discussed in Section 4.2.3 for the case of missing context information at the broker, this effect can be mitigated by maintaining old channel assignments in addition to the new ones for some time after a transition occurred. Such a mechanism would already be part of a more sophisticated self-adaptive version of eGRID.

*Benefit from
self-
adaptation*

As shown in this evaluation for the case of the channel-based scheme eGRID, self-transitions allow us to benefit from self-adaptability of filter schemes. Given the generic nature of self-transitions, we can easily integrate the potential of self-adaptations into an existing transition execution plan. This enables even more precise control over the performance and cost characteristics of the transition-enabled publish/subscribe system.

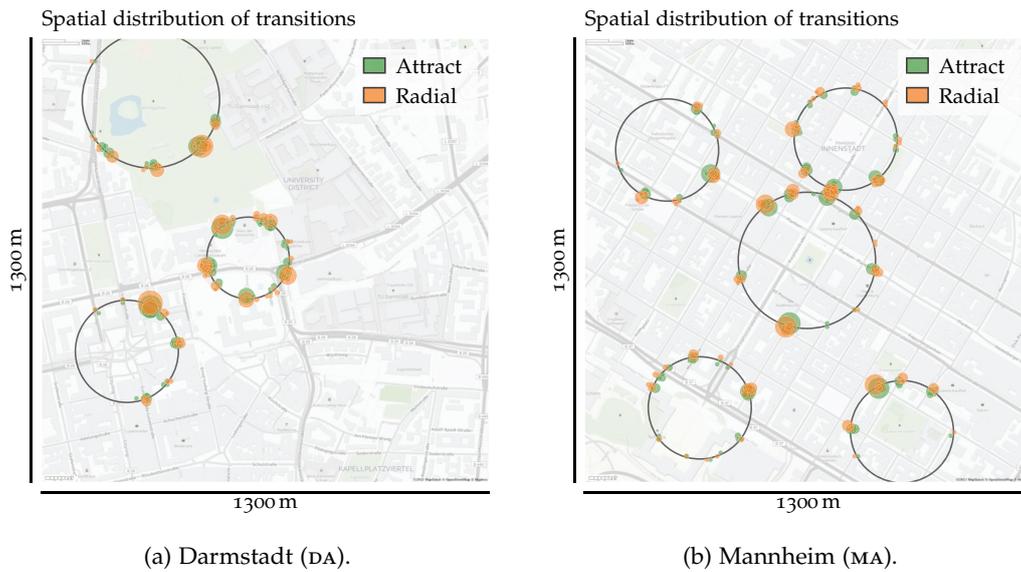


Figure 40: Locations of clients executing partial transitions between filter schemes.

6.2.4 Partial Transitions vs. Self-Adapting Mechanisms

By utilizing simple mechanisms and executing transitions between these mechanisms, we aim to achieve high adaptivity under dynamic environmental conditions. In the following evaluation scenario, we compare the performance and cost of filter scheme transitions to a self-adaptive filter scheme that offers the same functionality. Transitions are executed between the parametric scheme RADIAL and the channel-based scheme ATTRACT whenever a client approaches an attraction point.

The scenario is described in detail in Section 4.4 for coexisting transition-enabled mechanisms. However, in contrast to the execution plans discussed there, we only switch between the RADIAL and ATTRACT filter scheme. We neither use a local dissemination mechanism nor a gateway selection algorithm. Consequently, as soon as a client u approaches an attraction point and enters the area covered by a channel in ATTRACT, the partial transition $T^{\parallel}(u) : \text{RADIAL} \rightarrow \text{ATTRACT}$ is executed by the broker. Additionally, the client executes the total transition $T : \text{RADIAL} \rightarrow \text{ATTRACT}$. Accordingly, $T : \text{ATTRACT} \rightarrow \text{RADIAL}$ is executed by the client when leaving the area covered by the channel. In both cases, available state is transferred in-between the broker components of both filter schemes during the partial transition.

The behavior of the transition-enabled system is compared to that of MULTI, a filter scheme that realizes the same behavior within one mechanism. MULTI is an extended version of ATTRACT, where clients that are not assigned to one of the attraction points report their current location to the broker. The broker filters incoming events based on the reported location whenever a client is not assigned to a channel.

Figure 40 shows filter scheme transitions based on the client's current location when executing a transition. The radii of the attraction points within ATTRACT are

*Transitions
nearby
attraction
points*

*Self-adapting
mechanism in
comparison*

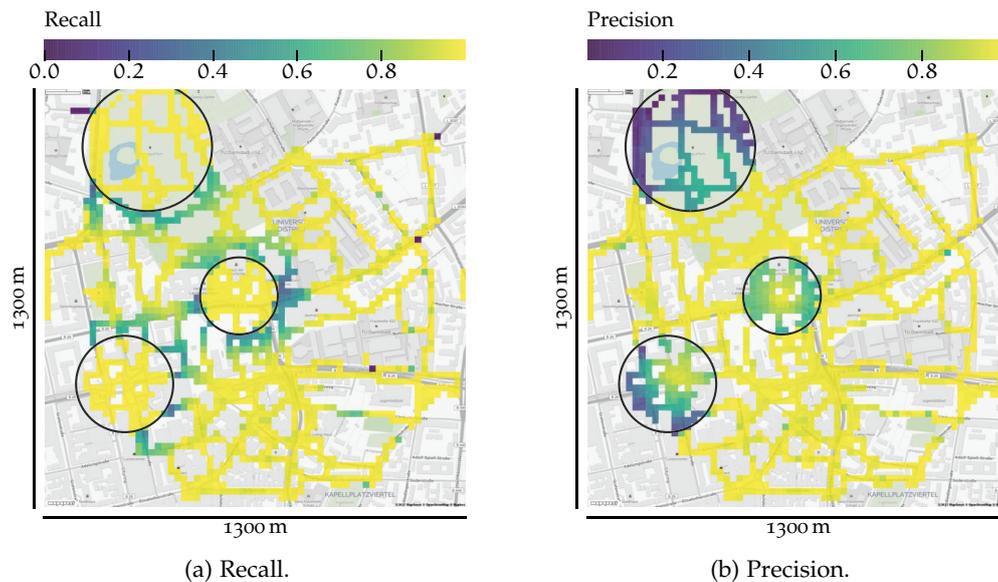


Figure 41: Performance of the transition-enabled system for the AR workload.

shown as black circles for reference. Each colored dish corresponds to a transition to the respective mechanism identified by the color. Transitions are aggregated for cells of 10×10 meters, with the radius of the circle indicating the number of transitions within the cell. As expected, transitions between RADIAL and ATTRACT are executed when clients enter or leave the area around the respective attraction point.

The resulting performance characteristics for the transition-enabled system depending on the target location of an event are shown in Figure 41 for the Darmstadt scenario. The recall within the area covered by channels in ATTRACT (indicated by the black circles) is approximately equal to 1.0. However, in direct vicinity of the respective area around an attraction point, the recall within the RADIAL scheme degrades significantly as shown in Figure 41a. Further away from any attraction points, RADIAL is again able to maintain a recall of 1.0. This behavior is caused by the fact that the radius of interest of a subscription is not taken into account by the channel-based scheme when assigning incoming location-based events to a channel. Consequently, all clients within a channel receive only events issued to that channel, regardless of the area covered by their original subscription. Thus, events created by clients right outside the area covered by the channel lead to the decreased recall.

The precision shown in Figure 41b reflects the characteristics of the individual filter schemes: while RADIAL allows precise filtering based on a client's current location, the channel-based scheme ATTRACT simply forwards an event to all clients within the channel. As a result, the precision depends on how similar the area covered by the channel is to the circular AoI given by the actual subscription of a client.

To also achieve high recall near the area covered by attraction points, we extend the ATTRACT scheme to take the radius of subscriptions into account when matching incoming location-based events. This is achieved by increasing the spatial extent of

Using the
subscription
radius

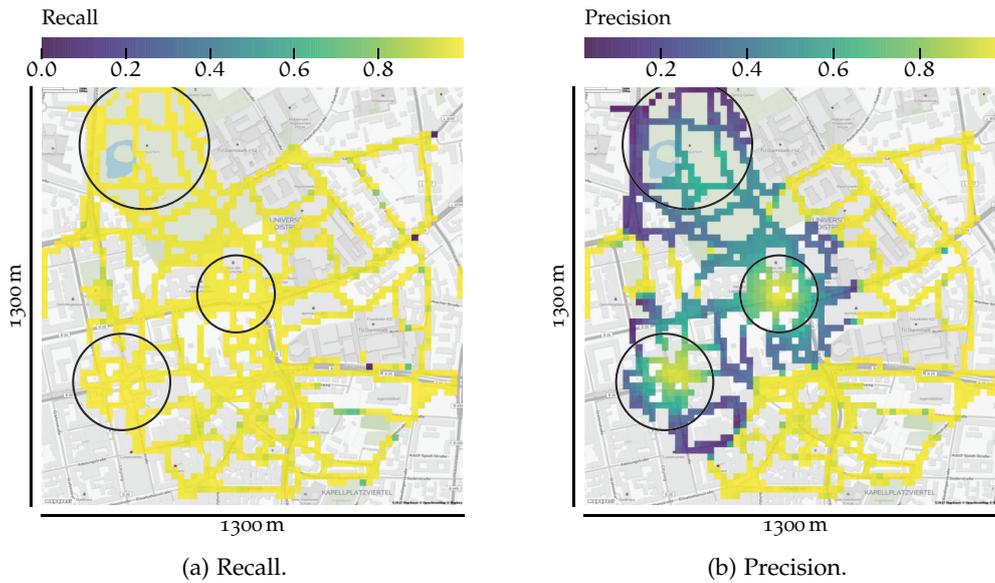


Figure 42: Extending channel coverage by taking the radius of subscriptions into account.

the channel with the maximum radius of interest out of the stored subscriptions. The resulting performance in terms of recall and precision is shown in Figure 42. We no longer observe lower recall around areas covered by a channel in ATTRACT, as shown in Figure 42a. Instead, the recall remains constantly high for all areas on the map. This is achieved at the cost of lower precision, as shown in Figure 42b. Events that are published within the additional area defined by the radius of interest of subscriptions are now forwarded to all subscribers of the channel as well.

For easier comparison, we show the aggregated performance and cost of both variants in Figure 43. As the effect does not only affect the transition-enabled system, but also the adaptive filter scheme MULTI, we report the respective results for both filter schemes with and without taking the radius of interest into account. As expected, both systems exhibit the exact same characteristics, showing that the functionality of a self-adaptive filter scheme can be realized using partial transitions between two less complex filter schemes. Additionally, this comes at virtually no additional costs in terms of traffic or decreased performance.

Taking the subscription radius into account improves the overall recall of the system in both cases as shown in Figure 43. The decrease in precision leads to additional cellular traffic caused by the transmission of unnecessary events for clients in the respective channels. This only affects clients that are within the area covered by a channel in ATTRACT, leading to the skewed distribution observable in Figure 43c.

The radius associated to a location-based subscription depends on the application at hand. This radius is constant for all mobile clients in our scenario, leading to a rather simple determination of the extension area required to achieve a recall of 1.0. If equal radii of interest cannot be assumed, one has to choose an appropriate extension area that achieves the desired trade-off between achievable recall and additional

*Trade-off
between recall
and precision*

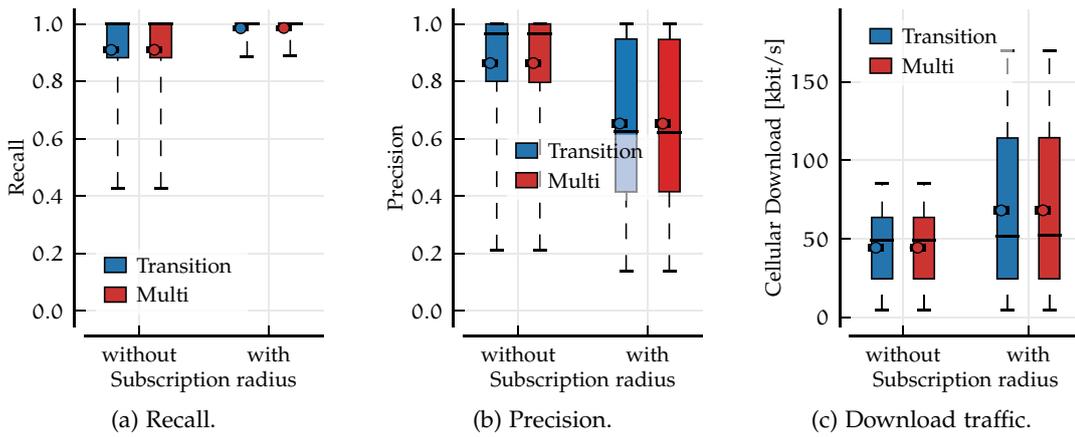


Figure 43: Performance and cost comparison.

traffic caused by the decrease in precision. To guarantee a recall of 1.0, the extension of the channel area has to match the maximum radius out of all subscriptions by clients within the respective channel. One can easily calculate this value based on the set of active subscriptions. However, each client entering or leaving the area could potentially alter the current maximum value, requiring frequent recalculation. Given the application-specific nature of attraction points, we believe that it is reasonable to assume that a suitable radius for such extension areas could be defined by the application beforehand.

We only reported results for the Darmstadt scenario in this section, as they match the behavior of the system in other conditions. For completeness, we report additional results for the Mannheim scenario and other workload models in A.2.

In this section of our evaluation we demonstrate the applicability of mechanism transitions to location-based filter schemes for publish/subscribe systems. We evaluate three distinct types of transitions as introduced in Chapter 4: *i) total transitions* affecting all clients of a broker, *ii) self-transitions* used for generic adaptation of a mechanism's state, and *iii) partial transitions* that allow parallel operation of multiple filter schemes for different groups of clients. Transitions between filter schemes enable the publish/subscribe system to adapt the performance vs. cost trade-off depending on the current application requirements or environmental conditions. We specifically study the impact of our proposed state transfer mechanism on the execution of transitions, showing its significance in achieving seamless transitions without degradations in performance. Partial transitions enable the utilization of application-specific knowledge by switching filter schemes, for example, for clients that approach an attraction point. Combined with the evaluation of transition-enabled dissemination mechanisms in the following section, these insights form the basis for the evaluation of coexisting mechanisms presented in Section 6.4.

6.3 TRANSITIONS BETWEEN EVENT DISSEMINATION MECHANISMS

In addition to filter schemes for location-based filtering, `BYPASS.KOM` includes mechanisms for locality-aware event dissemination. Depending on their configuration and the execution of transitions, `BYPASS.KOM` can be used in a pure ad hoc fashion or in a hybrid mode that relies on a cloud-based broker. In the following section, we assess achievable savings in terms of offloading the cellular connection by performing selected aspects of the brokering process directly on mobile clients. Therefore, we evaluate individual ad hoc dissemination mechanisms and different modes of hybrid operation as proposed in Section 4.3.2.

In Section 6.3.3 we conduct a proof-of-concept evaluation of centrally coordinated transitions between dissemination mechanisms. The central coordination of transitions between local dissemination mechanisms might lead to unnecessary overhead, given that clients already communicate locally. Therefore, in Section 6.3.4, we evaluate the utilization of the self-healing mechanism presented in Section 4.3.3 to spread transitions within an ad hoc network. Understanding both concepts of coordinating transitions between dissemination mechanisms lays the foundation for the evaluation of coexisting transition-enabled mechanisms in Section 6.4.

6.3.1 *Pure Ad Hoc Event Dissemination*

Before evaluating the performance of the proposed hybrid dissemination modes, we first characterize the individual ad hoc dissemination mechanisms in `BYPASS.KOM`. Their performance is evaluated within the Darmstadt scenario, and aggregated results for the `AR` and `RND` workloads are shown in Figure 44. While `RND` publishes events to arbitrary locations, the `AR` workload only issues local events, allowing the dissemination mechanism to utilize geofencing to limit the spread of messages. Additionally, events in the `AR` workload only need to reach clients within vicinity of the publisher to achieve a high recall, while in `RND` they need to reach the respective random target location. This location might only be reachable via multiple intermediate hops or it might not be reachable at all if network partitions occur. In addition, as messages are not subject to geofencing, the overall load on the network increases significantly for `RND` compared to the `AR` workload.

The results reported in Figure 44 support this rough assessment regardless of the respective dissemination mechanism. In general, the average achievable recall for pure ad hoc dissemination in the `RND` workload lies well below 0.4, with `Gossip` and `HyperG` achieving an average recall of below 0.2, as reported in Figure 44a. The single-hop dissemination mechanism `Broadcast` is obviously not suited to disseminate events to arbitrary target locations, with an average of 1% of all events reaching their intended receivers. The more aggressive schemes `Flooding` and `Plan-B` perform best for random target locations, yet reaching only approximately 35% of all subscribers on average.

If events are only to be disseminated to nearby mobile clients (as is the case in the `AR` workload) all multi-hop ad hoc dissemination mechanisms perform reasonably

*Impact of
geofencing*

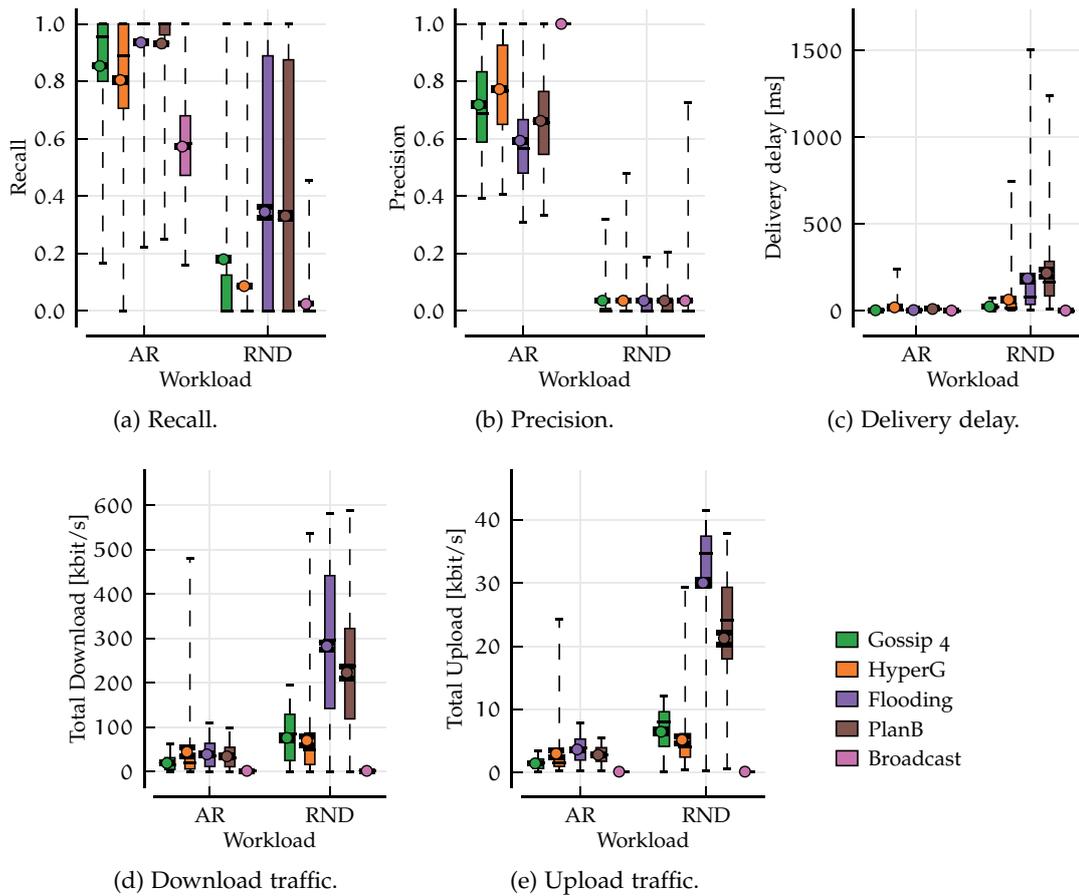


Figure 44: Performance and cost of pure ad hoc dissemination.

well. While Gossip and HyperG achieve an average recall of 0.85 and 0.8, respectively, with Flooding and Plan-B approximately 95 % of all subscribers are notified. The precision shown in Figure 44b illustrates the impact of geofencing within the AR workload. As events in the RND workload essentially need to be spread to every mobile client, the resulting precision is expected to be low, even in the face of a generally low recall. With geofencing, only clients that are actually subscribed to an event that is being disseminated forward it, leading to an average precision of between 0.6 and 0.8 for the multi-hop dissemination mechanisms. The high precision of the single-hop mechanism Broadcast for the AR workload is expected, given that events only reach clients within direct vicinity of the producer.

The delivery delay of ad hoc dissemination mechanisms shown in Figure 44c is generally in the order of a few milliseconds, provided that the network is not overloaded. However, once the network is overloaded (as is the case in the RND workload for HyperG, Flooding, and Plan-B) the delay increases significantly. At the same time, the probability of collisions during the wireless transmission increases, leading to the aforementioned drop in achievable recall. Figure 44d shows the download traffic per client for both workloads. The effect of geofencing is clearly visible for the

Delay and traffic characteristics

multi-hop dissemination mechanisms. On average, the download traffic of mobile clients increases by a factor of six if the spread of messages is no longer limited by geofencing. The same effect can also be observed for the average upload traffic for each mobile client. Here, the aggressive forwarding policies of FLOODING and PLAN-B result in a significant increase in upload traffic compared to the less aggressive mechanisms GOSSIP and HYPERG.

The performance of an ad hoc dissemination mechanism depends on the availability of sufficient forwarders within proximity. Therefore, we also study the recall as a location-dependent metric for the individual dissemination mechanisms. The results for the AR workload are shown in Figure 45. Plots show the average recall for events based on their target location, resulting in measurements at pedestrian walkways for the AR workload.

Location-dependent recall

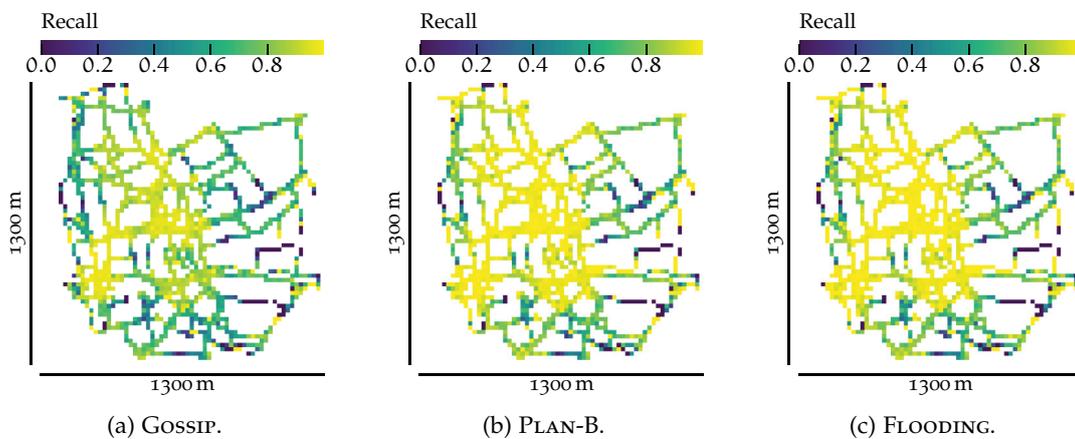


Figure 45: Recall depending on the target location of an event, AR workload.

All mechanisms perform better in areas with higher client density compared to more remote areas with lower client density on the map.⁴ Also, as discussed previously, PLAN-B and FLOODING achieve a higher recall due to their more aggressive transmission strategy compared to Gossip. However, even aggressive mechanisms are unable to maintain their high recall at more remote areas of the map. This is even more significant within the RND workload, shown in Figure 46. Here, we cannot benefit from geofencing when disseminating events, leading to a general increase in the number of messages and, thus, collisions in the network, as previously discussed. Still, the effect of client density on the achievable recall is clearly visible for all dissemination mechanisms.

For comparison, we also report the results for the RND workload with the random waypoint mobility model in Figure 47. The average client density is too low for Gossip with a forwarding probability of $p = 0.4$, leading to a generally low recall as shown in Figure 47a. FLOODING and PLAN-B benefit from their aggressive trans-

Random mobility for comparison

⁴ For a discussion of the client density resulting from the Darmstadt mobility model please refer to Section 6.1.1.

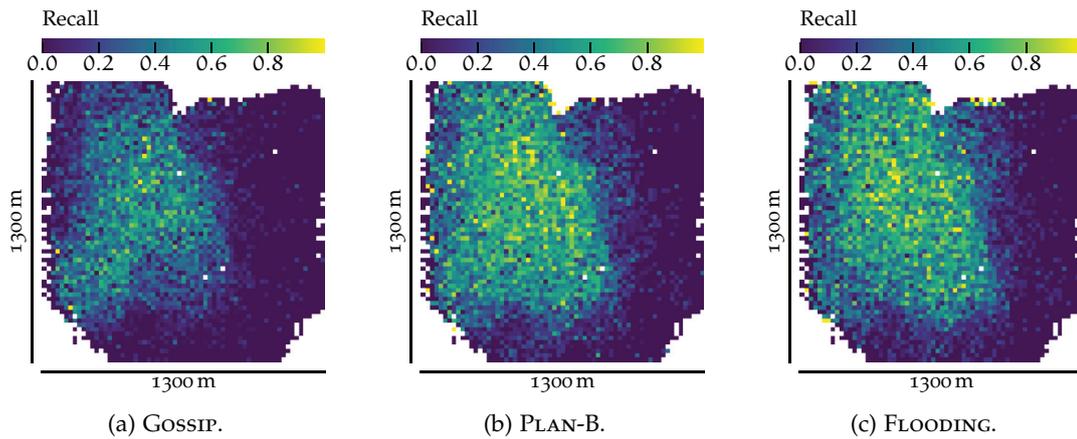


Figure 46: Recall depending on the target location of an event, RND workload.

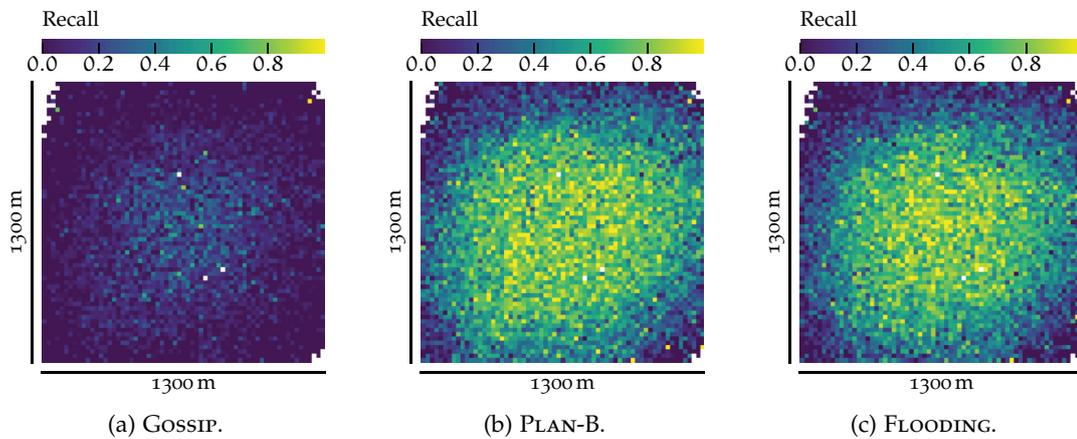


Figure 47: Recall achieved under the random waypoint mobility model, RND workload.

mission strategy and the rather homogeneous distribution of clients, achieving high recall in the central region of the simulated area.

The results presented in this section motivate the utilization of ad hoc dissemination mechanisms for transmission over a few hops and in rather dense scenarios. They also clearly show that pure ad hoc dissemination of events is not suitable, especially if events are not only relevant near their producer. Within *BYPASS.KOM*, we therefore consider two distinct application areas for ad hoc dissemination of events: *i) hybrid operation modes* as evaluated in the following section, and *ii) dissemination around application-specific attraction points* as evaluated in Section 6.4.

6.3.2 Hybrid Dissemination Mechanisms

In this section, we discuss the cost and benefits of the hybrid dissemination modes introduced in Section 4.3.2 and [147] under different workload models. Understand-

ing the characteristics of hybrid event dissemination serves as the foundation for the following discussion of transitions between event dissemination mechanisms, especially with respect to their coordination.

The modes and their key differences with respect to the dissemination of events are summarized in Table 7. The three modes OFFLOAD, AUGMENT, and GATEWAY utilize direct ad hoc dissemination for local events. In our evaluations, we rely on the Gossip dissemination mechanism with geofencing enabled as described in Section 4.3.1. While in OFFLOAD only non-local events are sent to the cloud for further dissemination, the cloud is used for all events in the remaining two hybrid modes. When sending events to a group of clients, GATEWAY utilizes local gateways that notify their one-hop neighbors via broadcast (BROADCAST). For the GATEWAY mode, we utilize SKC_LEACH as gateway selection algorithm. AD HOC and CLOUD are included for reference, given that they represent the two extreme cases of purely local and purely cloud-based event dissemination.

MODE	AD HOC DISSEMINATION	CLOUD DISSEMINATION	GATEWAYS
OFFLOAD	local	non-local	no
AUGMENT	local	local & non-local	no
GATEWAY	local	local & non-local	yes
AD HOC	local & non-local	none	no
CLOUD	none	local & non-local	no

Table 7: Hybrid dissemination modes studied in the evaluation.

Figure 48 reports the performance metrics for the aforementioned modes within the Darmstadt mobility model. To assess the impact of local and non-local events, we report the results for the AR and the LBS model, as introduced in Section 6.1.2. Considering the recall of 0.2 for AD HOC as reported in Figure 48a, pure ad hoc dissemination of events is not suitable for the generic location-based workload LBS. The potentially large distance between producer and consumer of an event leads to an increased probability of intermediate message loss. Additionally, the MANET is not guaranteed to be fully connected at all times, as partitions might occur due to client mobility. Even if events are to be distributed solely to clients within vicinity (as is the case in the AR workload) the recall reaches only 0.85. Utilizing a more sophisticated (or aggressive) local dissemination mechanism can increase the recall in this case. However, this comes at the cost of increased communication overhead and traffic in the MANET, potentially increasing the number of collisions.

Low recall of pure ad hoc mode

The precision reported in Figure 48b supports the claim that a pure ad hoc dissemination is not suitable for our scenario. The geofencing mechanism for local events limits the spread of messages to the vicinity of the interested clients, thus leading to an average precision of 0.7 in the AR workload. For generic location-based events as issued in the LBS workload, a larger number of clients is notified on the path towards

Precision depending on geofencing

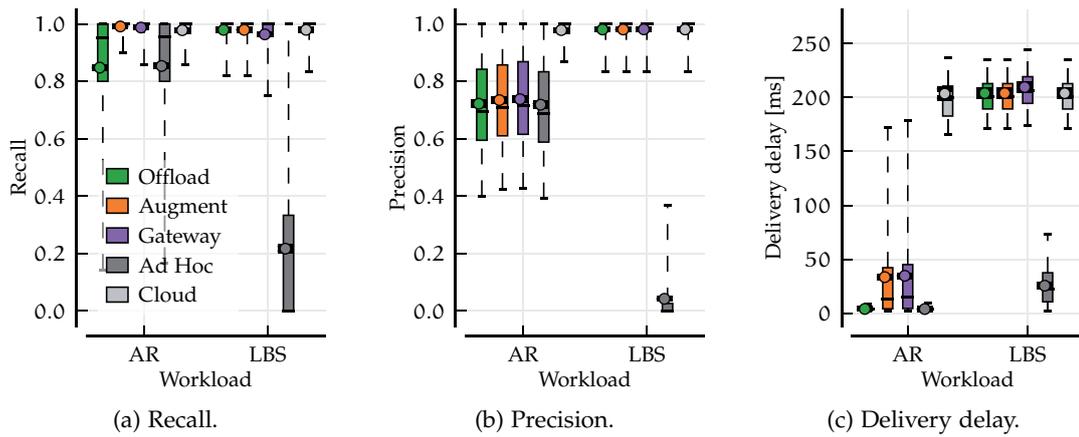


Figure 48: Performance of hybrid event dissemination mechanisms.

the intended group of receivers. This leads to the significant decrease in precision for the AD HOC dissemination mechanism.

*Low local
delivery delay*

Compared to a pure cloud-based dissemination mechanism, the AD HOC mode substantially speeds up the delivery of events. This effect is best visible if events are to be distributed only in the local neighborhood, leading to only a few milliseconds of average dissemination delay as reported in Figure 48c. In comparison, a pure cloud-based dissemination takes an average of 200 ms caused by the network delay between mobile user and cloud-based back end. Given a reliable cellular connection, recall and precision for CLOUD are nearly optimal for all workloads. The slight degradations in recall and precision for the cloud-based dissemination mode are caused by outdated state information of the underlying filter scheme. Clients report their current location only once every five seconds with the default configuration of the RADIAL filter scheme used in this evaluation. Therefore, events that are published right at the edges of a client’s AoI might not yet (or no longer) be covered by the AoI calculated at the broker based on the last reported location.

*Ad Hoc and
Cloud define
performance
bounds*

In the following, we discuss the hybrid modes of operation compared to the CLOUD and AD HOC operation mode. Here, the AD HOC mode serves as the lower bound in terms of achievable delivery delay, and the CLOUD mode as an upper bound in terms of recall and precision. As expected, the OFFLOAD mode behaves as AD HOC in case there are only local events, as these events are not sent to the cloud. Consequently, it behaves like CLOUD for the LBS workload, as all non-local events are sent to the cloud without any ad hoc dissemination being utilized.

*Augmenta-
tion leads to
higher recall*

The AUGMENT mode achieves an even higher recall than CLOUD in case of the AR workload. This is due to the fact that local dissemination is not affected by the previously discussed problem of outdated state information at the broker. Instead, affected clients still receive the event via the ad hoc dissemination mechanism in most cases, slightly increasing the recall. Given that not only interested, but also some uninterested clients receive the locally disseminated event, the precision of AUGMENT for the AR workload resembles that of the pure ad hoc mode. For the LBS workload,

we cannot distinguish between AUGMENT and OFFLOAD, as both modes differ only in how they handle local events.

When utilizing gateways to distribute events to a group of local clients, there is a slight decrease in the observed recall compared to the AUGMENT mode. This, again, is caused by outdated state information during calculation of a suitable set of gateways at the broker, as the selected gateways might not be able to reach all interested clients. We further explore this effect in Section 6.4 for coexisting transition-enabled mechanisms. Although the GATEWAY mode does not affect the performance metrics reported in Figure 48, it has a significant impact on the traffic characteristics.

Offloading the cellular network with gateways

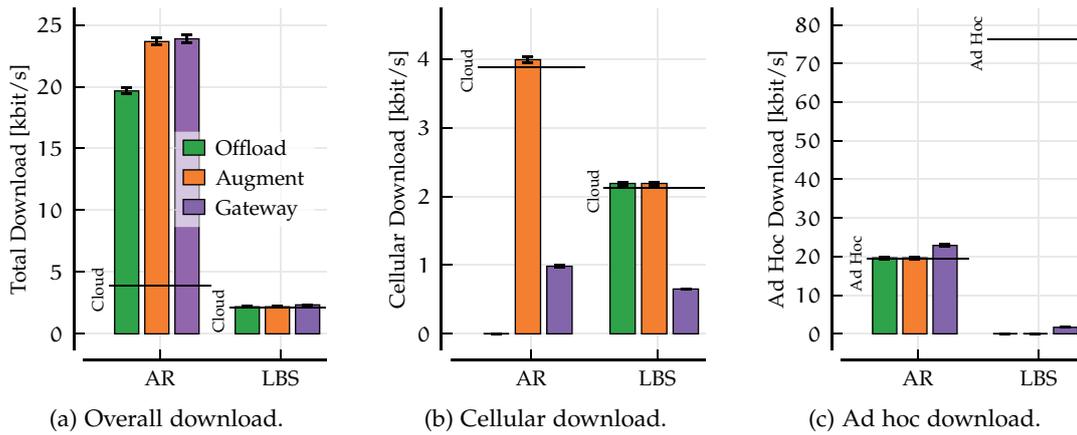


Figure 49: Per-client download traffic caused by hybrid dissemination mechanisms.

Figure 49 shows the traffic caused for clients by each of the hybrid modes, compared to the respective baseline. Considering the cellular download traffic as shown in Figure 49b, the pure cloud-based dissemination serves as the upper bound in cellular traffic, as indicated by the solid lines showing the average value.⁵ As expected, OFFLOAD and AUGMENT follow the traffic characteristics of CLOUD and AD HOC depending on whether local or non-local events are disseminated. With the utilization of gateways in the hybrid mode GATEWAY, the cellular download traffic per client is reduced to about 25% compared to CLOUD. As discussed previously, this reduction in cellular download traffic has no significant impact on the achieved recall for both workloads. At the same time, the addition of gateways only causes a slight increase in the ad hoc download traffic, as shown in Figure 49c.

Utilizing gateways in the event dissemination process contributes to the goal of locality-aware dissemination, as the gateways essentially act as local brokers for their direct vicinity. Their utilization avoids redundant transmissions of events to a potentially large number of clients within close proximity. Consequently, gateways

⁵ Actually, the hybrid configurations exhibit a slightly higher cellular download traffic than the pure cloud-based dissemination method. This is caused by additional information included in the message header. In all hybrid schemes, clients report two IP addresses to the cloud: one used by the cellular interface and the other used for local ad hoc communication. This additional information slightly increases the resulting message size. The effect is only visible due to the fact that the application payload is rather small, as defined in Section 6.1.2.

constitute a way to utilize locality of interest within the process of event brokering. We analyze the impact of gateways in more detail in Section 6.4.2 in the context of coexisting mechanisms. There, we also discuss the fairness characteristics of different gateway selection algorithms utilized within our work.

Based on our evaluation of individual ad hoc dissemination mechanisms and hybrid dissemination modes, we evaluate the execution of transitions between different dissemination mechanisms in the following section. Executing transitions between dissemination mechanisms enables us to later combine the selection of suitable gateways with ad hoc dissemination mechanisms in a locality-aware fashion.

6.3.3 Globally Coordinated Transitions

Given the cloud entity in any of the hybrid dissemination modes, our first approach to transitions between dissemination schemes relies on central coordination. As discussed previously, this is realized by disseminating transition execution plans to all affected clients via the cellular network. We focus on the execution of transitions and its impact on the overall system characteristics, addressing the last step of the MAPE-cycle introduced in Section 2.4. The analysis and planning steps are not within the scope of our work. To study the execution of transitions, we therefore trigger transitions at predefined times, as summarized in Table 8.

#	TIME	EXECUTION PLAN
T1	8 min	[T : GOSSIP → PLAN-B]
T2	12 min	[T : PLAN-B → HYPERG]
T3	17 min	[T : HYPERG → FLOODING]
T4	22 min	[T : FLOODING → BROADCAST]

Table 8: Execution plans for transitions between dissemination mechanisms.

*Recall and
precision over
time*

Figure 50 shows the resulting recall and precision of the transition-enabled system over time for the AR workload. For comparison, the average values for each individual mechanism are included. As for the case of transition-enabled filter schemes discussed previously, the transition-enabled system adapts its characteristics to the individual mechanism after each transition rapidly. As these characteristics are discussed in detail in Section 6.3.1, we instead focus on the timing of transitions and the system's behavior during their execution in the following. Given the bad performance of pure ad hoc event dissemination without geofencing, we only discuss the transition-enabled system under the AR workload in this section. For a discussion of the performance and cost of local dissemination mechanisms under other workload and mobility models, please refer to A.3.

In contrast to the case of filter schemes, we do not transfer any state information between individual dissemination mechanisms. Additionally, incoming messages can

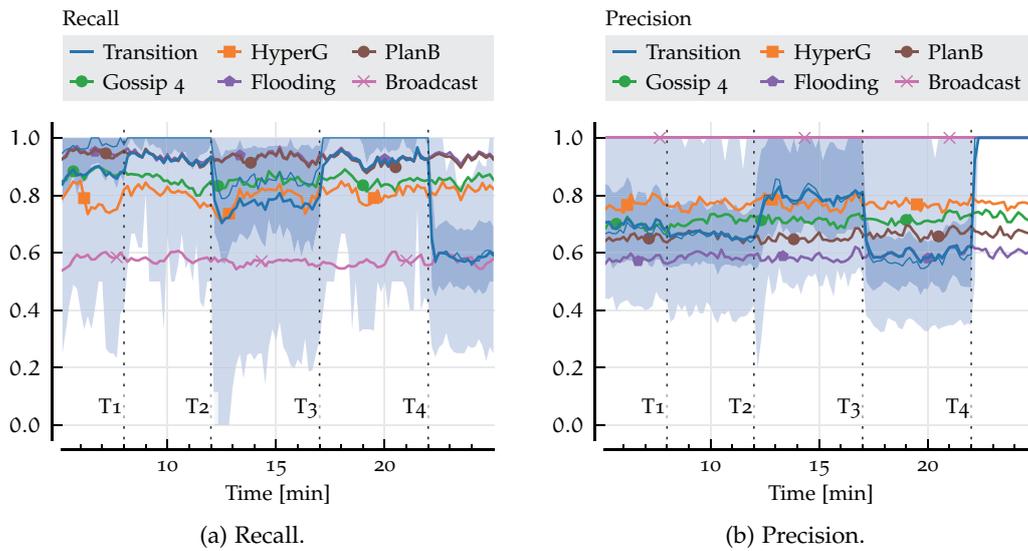


Figure 50: Recall and precision of transition-enabled dissemination mechanisms.

only be processed by a client if they were created with the same dissemination mechanism, potentially leading to message loss during a transition. As shown in Figure 51b, all clients within the given scenario execute the transitions within a time frame of 50 ms. The plot indicates the location of a client when executing a transition and the duration measured from the point in time where the first instance of this transition is being executed in the network. This time frame is given by the configuration of the cellular network and its latency as introduced in Section 6.1.3.

Timing of the transition

Given that all clients receive the transition trigger within a rather confined time frame, the effect of message loss due to incompatible dissemination mechanisms is negligible. This is shown in Figure 51a, where the recall is plotted during all four transitions. Although the effect of message loss can be observed for the transition T1, where the recall drops for approximately 50 ms, it is neither significant nor clearly visible for the remaining transitions. The drop for transition T4 at around 250 ms after the transition is executed is not caused by the transition itself but rather by the new dissemination mechanism: the one-hop dissemination via BROADCAST achieves an average recall of 0.6 only, as already shown in Figure 50a.

Performance during transitions

As later discussed in Section 6.4, central coordination of transitions between dissemination mechanisms can cause significant control overhead. Additionally, in some scenarios, a central coordinator might not be available or network connectivity to all mobile clients cannot be guaranteed, as discussed in our work on pure ad hoc dissemination in [149]. In the following section, we study how the self-healing mechanism introduced in Section 4.3.3 can be utilized to spread a transition via an ad hoc network at no additional cost and without relying on central coordination.

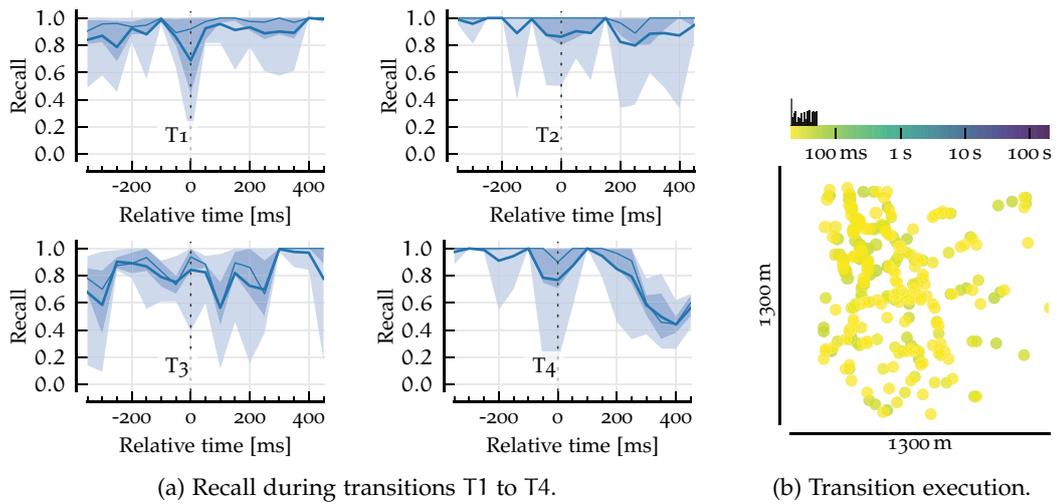


Figure 51: Recall before and after a transition execution.

6.3.4 Locally Initiated Transitions with Self-Healing

The self-healing mechanism introduced in Section 4.3.3 can also be utilized to spread a transition decision within a MANET. In contrast to centralized coordination as discussed in the previous section, this process significantly reduces cellular traffic, as evaluated in Section 6.4.3 within the context of coexisting mechanisms. In this section, we characterize the spread of transitions in a MANET when relying solely on the self-healing mechanism. It is important to keep in mind that the self-healing mechanism is *passive* in that it does not spread the transition decision actively to neighbors. Instead, it relies on piggybacking to include information about the currently utilized mechanism within outgoing messages. Consequently, the transition decision is spread as a side-effect of the message forwarding procedure.

In this evaluation, transitions are again triggered at predefined times as previously discussed and summarized in Table 8. Instead of distributing the transition decision to all mobile clients via the cellular network, only a subset of clients is notified. The transition decision then spreads via the self-healing mechanism whenever a client forwards a message. Figure 52 shows the recall and precision over time when transitions are triggered on a single randomly chosen client. For comparison, the behavior resulting from global coordination as discussed in the previous section is included in the plots. While the overall behavior is very much aligned to the case of global coordination, the effect of a transition is delayed. This effect is clearly visible after transitions T1 and T4 in Figure 52a.

To study this effect in more detail, we plot the currently utilized mechanism over time for each client in Figure 53. In Figure 53a, each horizontal line corresponds to one single client. Clearly, most clients execute the transition at roughly the same time. However, a fraction of clients executes some of the transitions very late or not at all. For better readability, Figure 53b shows the same data but grouped by mechanism.

Triggering a single client

Per-client mechanism utilization

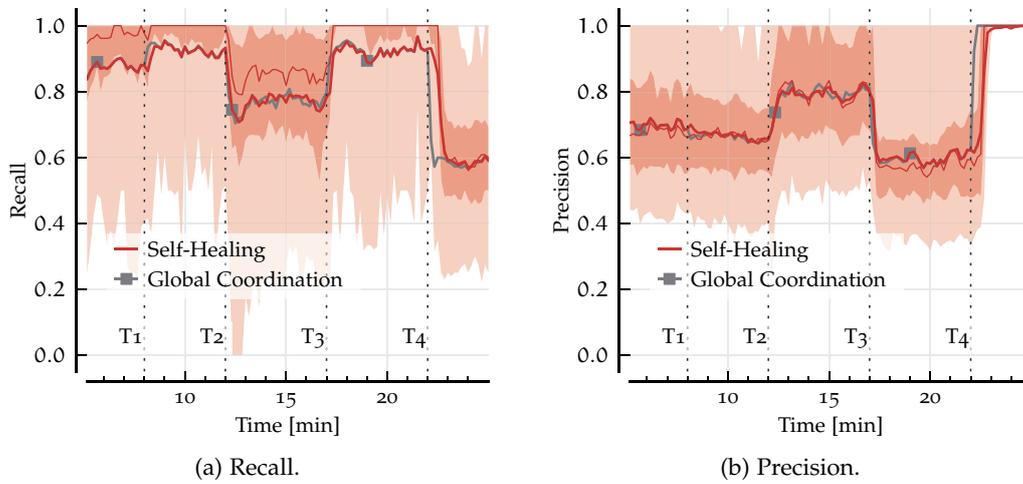


Figure 52: Self-healing compared to centrally coordinated transitions.

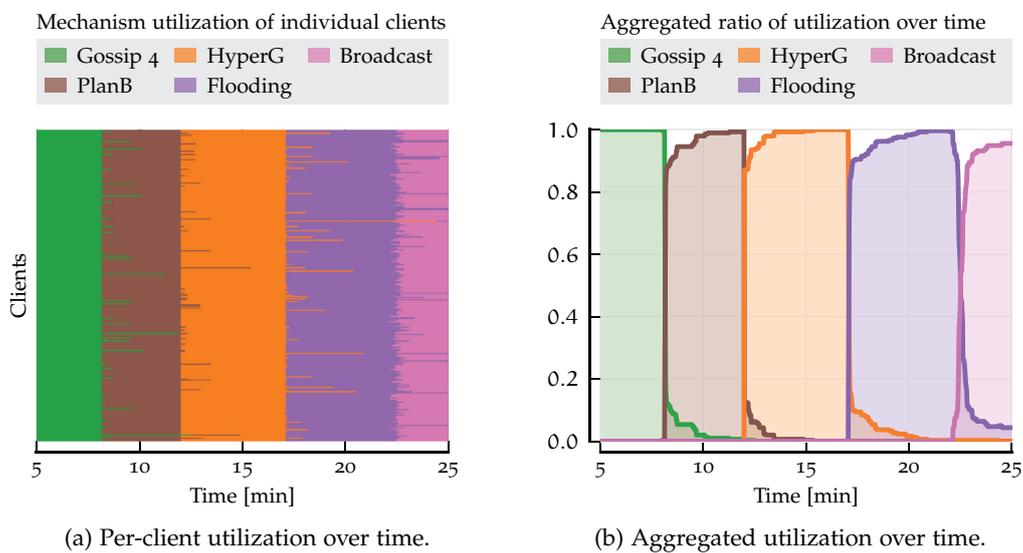


Figure 53: Mechanism utilization in case of transitions spreading via self-healing.

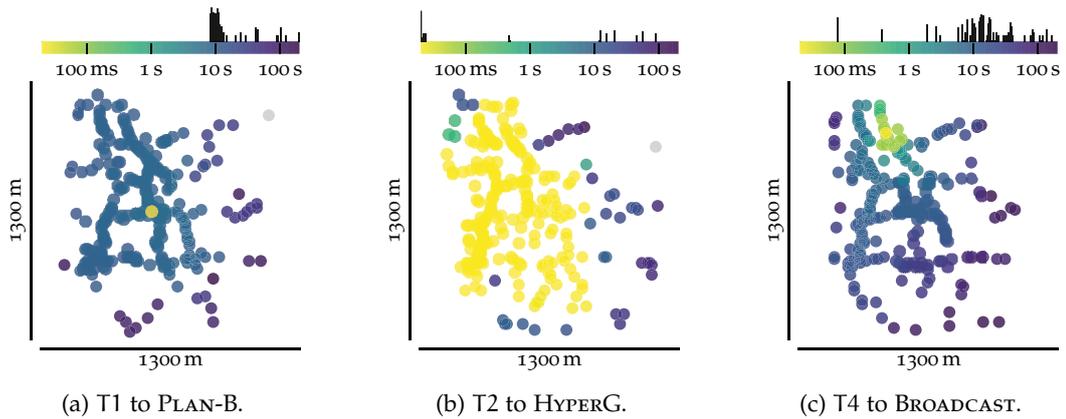


Figure 54: Geographical spread of transitions with one initiator.

Here, it is clearly visible how a transition affects 90 % of the mobile clients on average, with the remaining clients taking significantly longer to execute the transition. This effect is especially severe for transition T4 to BROADCAST, as this dissemination mechanism does not forward messages more than one hop. Consequently, the transition decision induced by a single client at minute 22 has still not reached all clients at the end of the observation three minutes later.

Geographical spread of transitions

To understand this behavior, we need to take the physical location of clients into account. In Figure 54 we therefore plot the location of clients when executing a transition. The color indicates the delay between initiation and execution of a transition for each mobile client. A histogram of the respective delays is included for a better assessment of the characteristics of each individual transition. We can distinguish between three cases: *i*) high initial delay and rapid spread (Figure 54a), *ii*) low initial delay and rapid spread (Figure 54b), and *iii*) slow spread (Figure 54c). The reasons for the respective behavior are explained in the following.

High initial delay, rapid spread

For transition T1 to PLAN-B shown in Figure 54a, we observe that only the initially triggered client in the center of the simulated area executes the transition immediately. However, after a delay of approximately ten seconds, the decision is rapidly spread to a large fraction of the remaining clients. Only a few clients in the outskirts of the simulated area do not receive the respective information, leading to increased delay of approximately one to two minutes. This is due to the fact that PLAN-B utilizes a distance-based hesitation mechanism, as discussed in Section 4.3.1 and [68]. Consequently, a client located within a rather densely populated area might not forward any messages at all. In this case, the transition decision can only spread if the client itself initiates a new message transmission, which can lead to a high initial delay, as observed in Figure 54a. However, once the respective message is sent, it is rapidly spread to a large fraction of clients.

Low initial delay, rapid spread

When executing transition T2 to HYPERG, one can observe a similar behavior, although this time the initial delay until further clients execute the transition is very low. This is caused by the active exchange of Hello messages in HYPERG [86] to

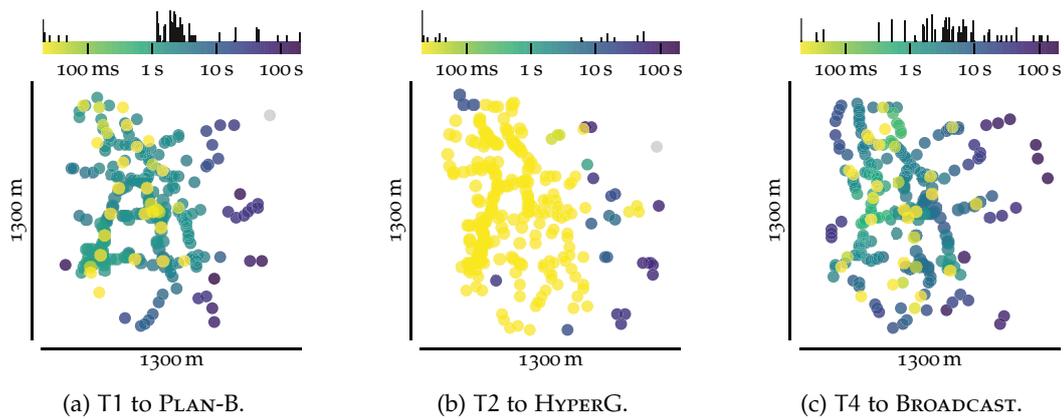


Figure 55: Geographical spread of transitions with 30 initiators.

determine the density of the current neighborhood. These messages are broadcast every 100 ms. As every outgoing mechanism-specific message, they contain piggy-backed information used by the self-healing mechanism as discussed in Section 4.3.3. Consequently, a transition is spread rapidly to nearby clients regardless of whether application payload needs to be forwarded or not.

For a single-hop dissemination mechanism like BROADCAST the achievable spread is clearly limited, as shown in Figure 54c. As messages are not actively forwarded, a transition decision can only spread if a client actively publishes an event. Given that each broadcast reaches multiple clients, the probability of a client publishing an event after having executed the transition increases significantly after the initial broadcast took place. Consequently, as already discussed based on Figure 53b, a large fraction of clients executes the transition within approximately 10 to 20 seconds.

A high initial delay is not necessarily bad with respect to message loss, given that only the initiator is unable to process incoming messages at this stage. However, once the decision is spread to the remaining clients, it should reach a large fraction of the clients within a short time frame to prevent message loss. The previously discussed results for the recall shown in Figure 52a support this statement. While the behavior is slightly delayed, the recall does not drop compared to a configuration with global coordination of transitions.

In hybrid configurations, and especially when considering the utilization of gateways, a transition decision could be injected at multiple clients. While this slightly increases the load on the cellular network, it comes at the advantage of a more rapid spread of the decision to the remaining nodes. This is visible in Figure 55, where the transition decision is sent to 10% of the clients initially, corresponding to 30 clients in our scenario. While the overall behavior of the individual mechanisms remains as previously discussed, the average delay for the spread of the decision decreases significantly compared to the results reported in Figure 54. Results for FLOODING are omitted in this section for brevity, as the mechanism behaves similar to PLAN-B.

*Slow spread
for single-hop
mechanisms*

*Triggering
multiple
clients*

The results presented in this part of our evaluation motivate the utilization of hybrid modes for event dissemination. We show that pure ad hoc dissemination of events is not feasible within our scenario, given that the dissemination mechanisms are unable to cover the whole simulated area. Still, our results motivate the utilization of direct ad hoc dissemination for locally relevant events. The utilization of gateways further contributes to the goal of locality-aware event dissemination. By sending events only to a subset of clients within a region and then distributing them to the remaining clients in an ad hoc fashion, the cellular connection is offloaded significantly, while high recall is maintained.

Given that clients are not evenly distributed in our scenario, we want to utilize application-specific knowledge about attraction points, as already discussed for filter schemes. Our evaluation shows that transitions between dissemination mechanisms can be executed either in a centrally coordinated fashion or by relying on our self-healing mechanism. Regardless of the execution method, the performance is not negatively affected. This motivates the combined analysis of filter schemes and dissemination mechanisms and their transitions presented in the following section.

6.4 COEXISTENCE OF TRANSITION-ENABLED MECHANISMS

In this section, we characterize system configurations that rely on coexisting mechanisms and the respective transitions, as discussed in Section 4.4 and further detailed in Section 5.3.2. Consequently, we combine mechanisms for location-based filtering and mechanisms for locality-aware event dissemination by executing the respective transitions based on application-specific knowledge about attraction points. With respect to the utilized filter schemes, the behavior remains the same as reported in Section 6.2.4 for the utilization of partial transitions for clients near attraction points. However, we vary the utilization of dissemination mechanisms for direct local dissemination and gateways according to the execution plans detailed in Section 5.3.2.

6.4.1 *Direct Dissemination of Local Events near Attraction Points*

Transitions activating or deactivating a local dissemination mechanism are executed whenever a client enters or leaves the area covered by a channel in ATTRACT, as discussed in Section 5.3.2. This configuration of BYPASS.KOM is hereafter referred to as CX-AH. When activated, local dissemination is only used for *local events*, i. e., events that are published to a client's current location. Consequently, an effect should only be visible for the AR workload.

Figure 56 reports the performance characteristics of the CX-AH configuration. For comparison, we include a configuration where locally published events are *not* forwarded to the cloud. Instead, events are only distributed via the ad hoc protocol and the purely cloud-based event dissemination relying on RADIAL. In this configuration, we do not execute any transitions. The recall reported in Figure 56a corresponds to the results reported for hybrid event dissemination in Section 6.3.2. However, as local dissemination is limited to the area defined by attraction points, the effects are less

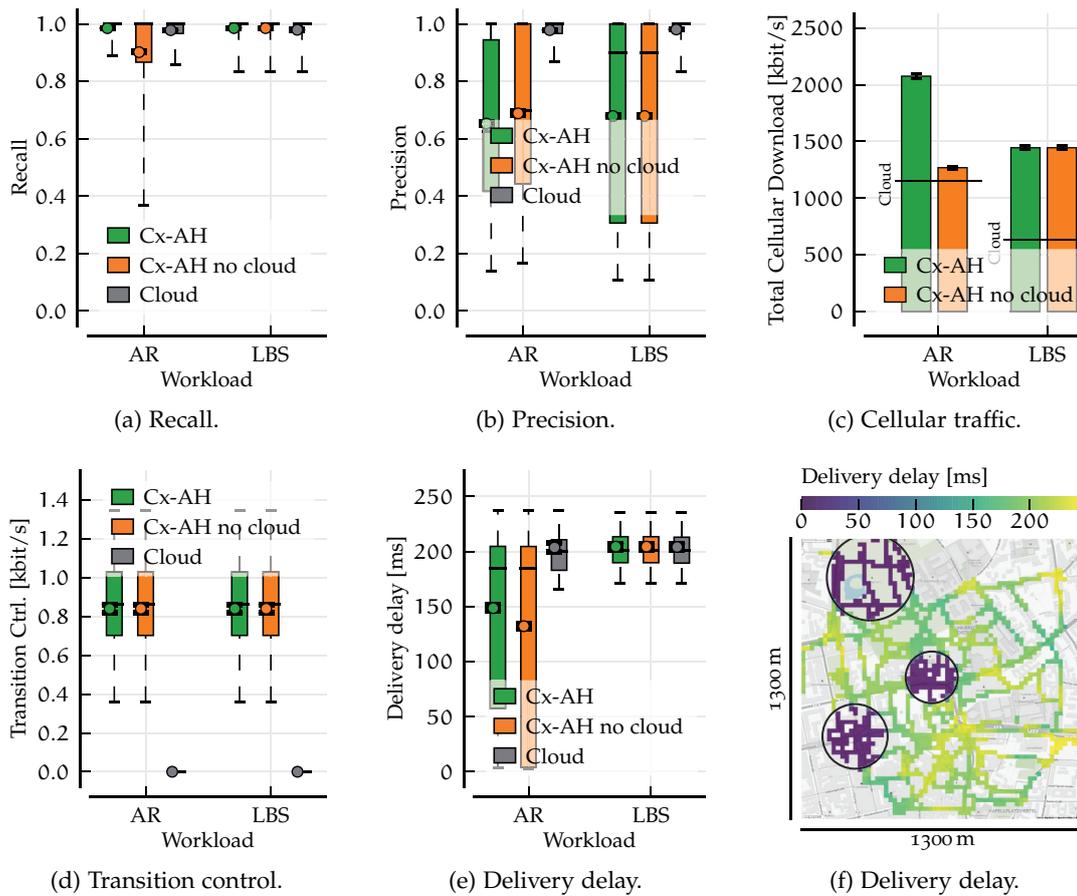


Figure 56: Performance of ad hoc dissemination around attraction points.

significant, as they only affect a subset of all mobile clients. As expected, there is no change in recall for the LBS workload, as clients do not issue any local events.

The precision reported in Figure 56b drops due to the utilization of the channel-based filter scheme *ATTRACT* for both workloads, as previously discussed in Section 6.2.4. Given that *CX-AH* behaves as the previously discussed hybrid configurations for clients within reach of an attraction point, we do not expect any offloading effect. This is supported by the results reported in Figure 56c: the overall cellular traffic increases due to the utilization of the channel-based filter scheme and its decreased precision.

To assess whether the increase in traffic compared to the purely cloud-based approach is caused by the filter scheme or by the coordination of transitions, we report the traffic caused by transition coordination in Figure 56d. Compared to the overall traffic reported in Figure 56c, the coordination of transitions accounts for less than 0.1% of the cellular traffic in both configurations of *CX-AH*. Still, the average delivery delay is decreased from approximately 200 ms to about 150 ms if ad hoc dissemination mechanisms are utilized nearby attraction points, as reported in Figure 56e.

The spatial distribution of the delivery delay is more interesting than average values, given that local dissemination mechanisms are only used nearby attraction points. Therefore, we report the delivery delay depending on the location of a client in the AR workload in Figure 56f. The figure shows that local ad hoc dissemination is limited to the channels within ATTRACT, with an average delivery delay of below 50 ms. For locations that are not covered by a channel the delivery delay remains at around 200 ms, as also reported for the cloud-only configuration.

6.4.2 Utilizing Gateways nearby Attraction Points

To offload the cellular connection, we utilize gateways in the area covered by a channel in ATTRACT, as discussed in Section 5.3.2. We utilize three distinct gateway selection algorithms (DKC-LEACH, CS-WCA, and SKC-LEACH) as representative examples of the algorithms included in the respective module published in [156]. For a more detailed discussion of the gateway selection algorithms, please refer to Section 4.3.4. As motivated, we evaluate different combinations of dissemination mechanisms and gateway selection algorithms, specifically focusing on the impact of multi-hop versus single-hop dissemination. We utilize GOSSIP with a forwarding probability of $p = 0.4$ as multi-hop dissemination mechanism, and UNICAST and BROADCAST as single-hop mechanisms. UNICAST can be utilized as the gateway selection procedure returns an assignment of clients to gateways, as discussed in Section 4.3.4. However, in this case, the contact information for the respective clients need to be forwarded to the gateway, leading to increased message sizes and, thus, additional cellular communication overhead. This effect is visible in Figure 57a when comparing the cellular traffic of the configurations using BROADCAST and UNICAST regardless of the utilized gateway selection algorithm. Figure 57a additionally shows that by utilizing gateways within ATTRACT channels, the cellular traffic is reduced by up to 64 % compared to a configuration without gateways.

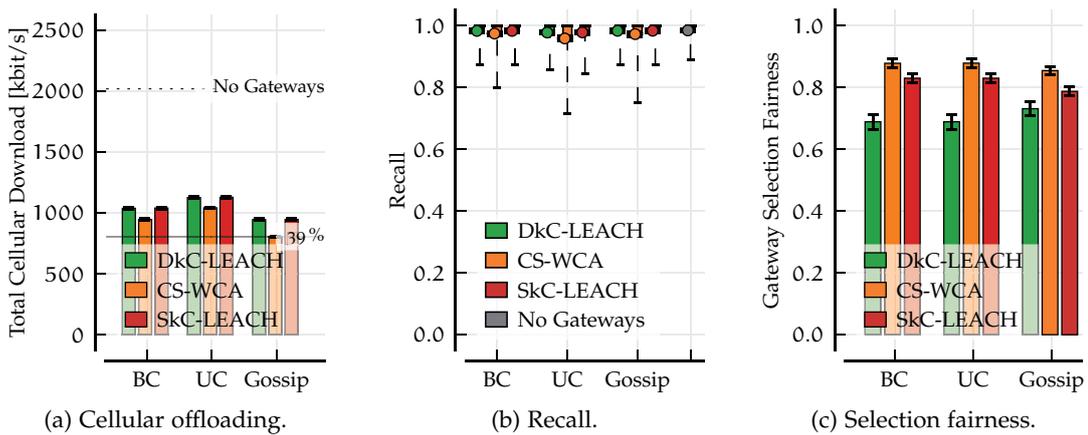


Figure 57: Achieved offloading, recall, and fairness characteristics (AR workload).

The chosen dissemination mechanism does not have a significant impact on the recall for `DKC-LEACH` and `SKC-LEACH`, shown in Figure 57b. The recall of `CS-WCA` is slightly worse compared to the other algorithms and a configuration without gateways. Before discussing the reason for this decrease, we briefly discuss the fairness of the individual gateway selection algorithms, shown in Figure 57c. The gateway selection fairness for all N clients is calculated using the fairness index defined by Jain in et al. [76]:

$$J(x) = \frac{\left(\sum_{i=1}^N x_i\right)^2}{N \sum_{i=1}^N x_i^2}.$$

Here, x_i is a counter of how often client i is chosen as a gateway. An algorithm with $J(x)$ closer to 1 distributes load across mobile clients more evenly.

The selection fairness of individual algorithms does not depend on the utilized dissemination scheme. The difference between the single-hop dissemination mechanisms and `GOSSIP` is caused by the configuration of the gateway selection algorithms: if a multi-hop dissemination mechanism is used, less gateways are requested by `BY-PASS.KOM`. This is controlled through the parameter k as introduced in Section 5.3.2. In `BY-PASS.KOM`, k is calculated based on the number of clients $|U|$ subscribed to the respective channel as $k = \lceil |U| \times k' \rceil$. For single-hop dissemination mechanisms we set $k' = 0.4$, corresponding to 40% of all subscribers being used as gateways. In case of multi-hop dissemination, k' is reduced to 0.2. Consequently, the selection fairness behaves slightly different, as less gateways are selected in this case. In general, given that we only consider how often a client acts as a gateway, the stochastic approaches `CS-WCA` and `SKC-LEACH` achieve slightly higher selection fairness.

In real-world applications, the selection fairness is usually not a suitable indicator to compare selection algorithms. Instead, one wants to consider the available battery power or the quality of the cellular network connection when determining suitable gateways. Here, more elaborate selection algorithms come into play, as discussed for offloading in transition-enabled monitoring systems in [156]. Figure 58 aids in understanding the previous results by showing spatial characteristics of the gateway selection algorithms. Figure 58a shows the average number of subscribers interested in an event as spatial metric, regardless of the utilized filter scheme. This corresponds to the ground truth resulting from the circular AoI defined by the application when issuing a location-based subscription.

In comparison, the number of utilized gateways for events issued at the respective locations is shown for `SKC-LEACH` in Figure 58b and `CS-WCA` in Figure 58c in case of single-hop dissemination via `BROADCAST`. The selection algorithm `CS-WCA` selects on average 20% less gateways compared to `SKC-LEACH`, especially for the large area of the top left attraction point. This observation is also backed by the aggregated number of utilized gateways per event in Figure 58d. Additionally, as a consequence of the transition execution plan discussed in Section 5.3.2, gateways are only utilized for events that are distributed to clients within a channel in `ATTRACT`. The area is larger than the radius of the individual attraction points, as we add the average subscription radius as previously discussed in Section 6.2.4.

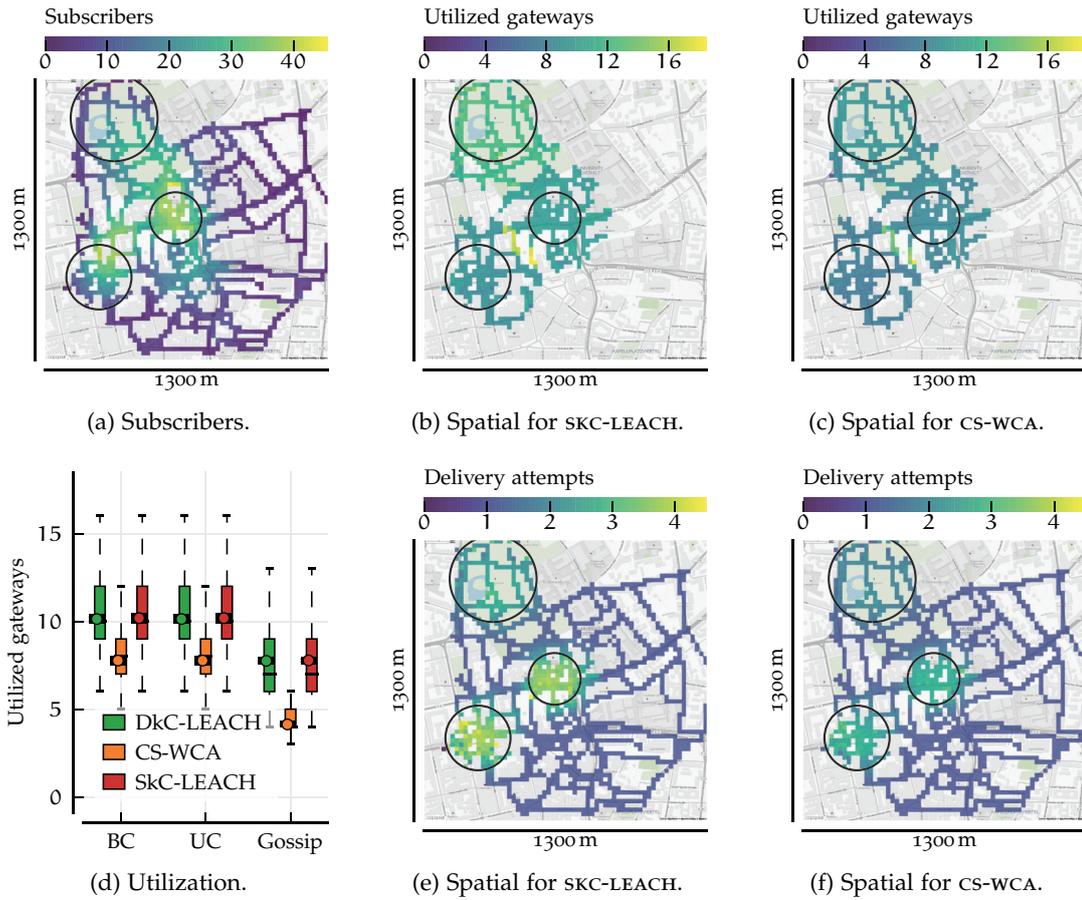


Figure 58: Utilization of gateways and delivery attempts (AR workload).

Figure 58e and Figure 58f report the number of delivery attempts for events based on their target location for SKC-LEACH and CS-WCA, respectively. A larger value indicates redundant coverage of a client by multiple gateways, as we utilize the single-hop BROADCAST dissemination mechanism. Due to the higher number of gateways being selected by SKC-LEACH, events are delivered up to four times to clients within proximity of an attraction point, as shown in Figure 58e.

This is due to the fact that we determine the preferred number of gateways (the parameter k) solely based on the number of subscribers and not on their average density. The clustering-based algorithm CS-WCA takes client density into account (as it has a direct impact on the size of a cluster), thereby reducing the number of gateways and, consequently, the delivery attempts compared to SKC-LEACH. A simple solution to include an estimate of the client density into the calculation of k is to utilize the radius r of an attraction point or to choose a gateway selection algorithm that already takes the node density into account. We propose an alternative approach in Section 5.3.2, relying on transitions to adapt the dissemination mechanism used by gateways based on the client density. The characteristics of this approach are discussed in the following section.

6.4.3 Executing Mechanism Transitions based on Client Density

As proposed in Section 5.3.2, execution plans for transitions can be determined based on the observed client density to further adapt the system to application-specific dynamics. In this section, we discuss the resulting characteristics of the system, focusing on the utilization of the respective mechanisms under varying client density. Consequently, this evaluation serves as a proof of concept for the applicability of our transition-specific contributions to the SIMONSTRATOR.KOM platform. It is not intended as a performance evaluation of the resulting system, given the rather arbitrary choice of our execution rules presented in Section 5.3.2.

We adapt the dissemination mechanisms used for the direct dissemination of local events and for gateways as discussed in Section 5.3.2. The respective transitions are executed if the observed client density ρ at an attraction point crosses the threshold δ . In this evaluation, we calculate ρ using the radius of an attraction point r in meters and the number of clients subscribed to the respective channel, $|U|$, as follows:

$$\rho = |U| \times \frac{10,000 \text{ m}^2}{r^2} .$$

Thereby, ρ normalizes the number of subscribers to the area of an attraction point with $r = 100 \text{ m}$. We report the system behavior for different values of δ .

In contrast to previous evaluations, we vary the total number of clients during the experiment to trigger the density changes. To this end, we utilize a churn model that starts with 100 clients. After 8 minutes of simulated time, the model linearly increases the number of active clients over a time span of 7 minutes, until finally reaching the total number of 297 active clients as in our default setup. The number of clients remains constant for another 3 minutes, before the model returns to 100

*Calculation of
 ρ and δ*

Churn model

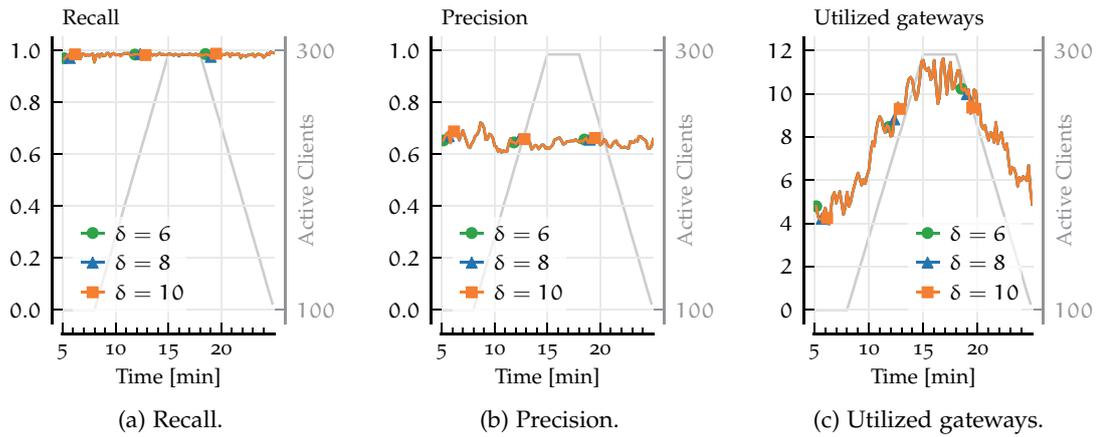


Figure 59: Performance over time for varying δ ; AR workload and DA scenario.

clients within a time span of 7 minutes. Regardless of whether a client is active or not, it moves according to the mobility models previously discussed.

Performance
over time

Figure 59 shows the performance characteristics of our system with varying δ over time. For reference, the number of currently active clients within the previously discussed churn model is included in each plot. The results for recall and precision show that the system behaves stable regardless of the configuration of δ or the execution of our proposed transitions. As expected, the number of gateways used to distribute an event increases with the density of clients, as more clients are active in the area around attraction points.

Transition
execution

To validate the correct operation of our execution plans, we plot the location of transitions executions affecting the gateway dissemination mechanism in Figure 60a and Figure 60b. In both cases, transitions are only executed on clients that are currently in the area assigned to an attraction point. For comparison, the locations of filter scheme transitions between *ATTRACT* and *RADIAL* are shown in Figure 60c and Figure 60d. Filter scheme transitions are only executed when clients enter or leave the area assigned to an attraction point. There is one exception: clients that join the system due to our churn model while being near an attraction point also execute the respective transition. This is visible in comparison to the results reported in Section 6.2.4 for a constant number of clients. Still, compared to transitions between gateway dissemination mechanisms, the execution of transitions between filter schemes mostly occurs at the border of the area assigned to an attraction point.

In the Mannheim scenario as shown in Figure 60b, clients near the attraction point on the top left of the map *exclusively* utilize *UNICAST* as gateway dissemination mechanism. The client density at this attraction point, ρ , never exceeded $\delta = 8$. Near all other attraction points, both mechanisms are utilized by clients. Figure 61 reports the corresponding mechanism utilization over time for different values of δ in the DA scenario. A ratio of 1.0 would indicate that all 297 clients are currently utilizing the respective mechanism. In all three cases, approximately 35% of all clients are subscribed to an attraction point channel. Depending on the value of δ , at some point

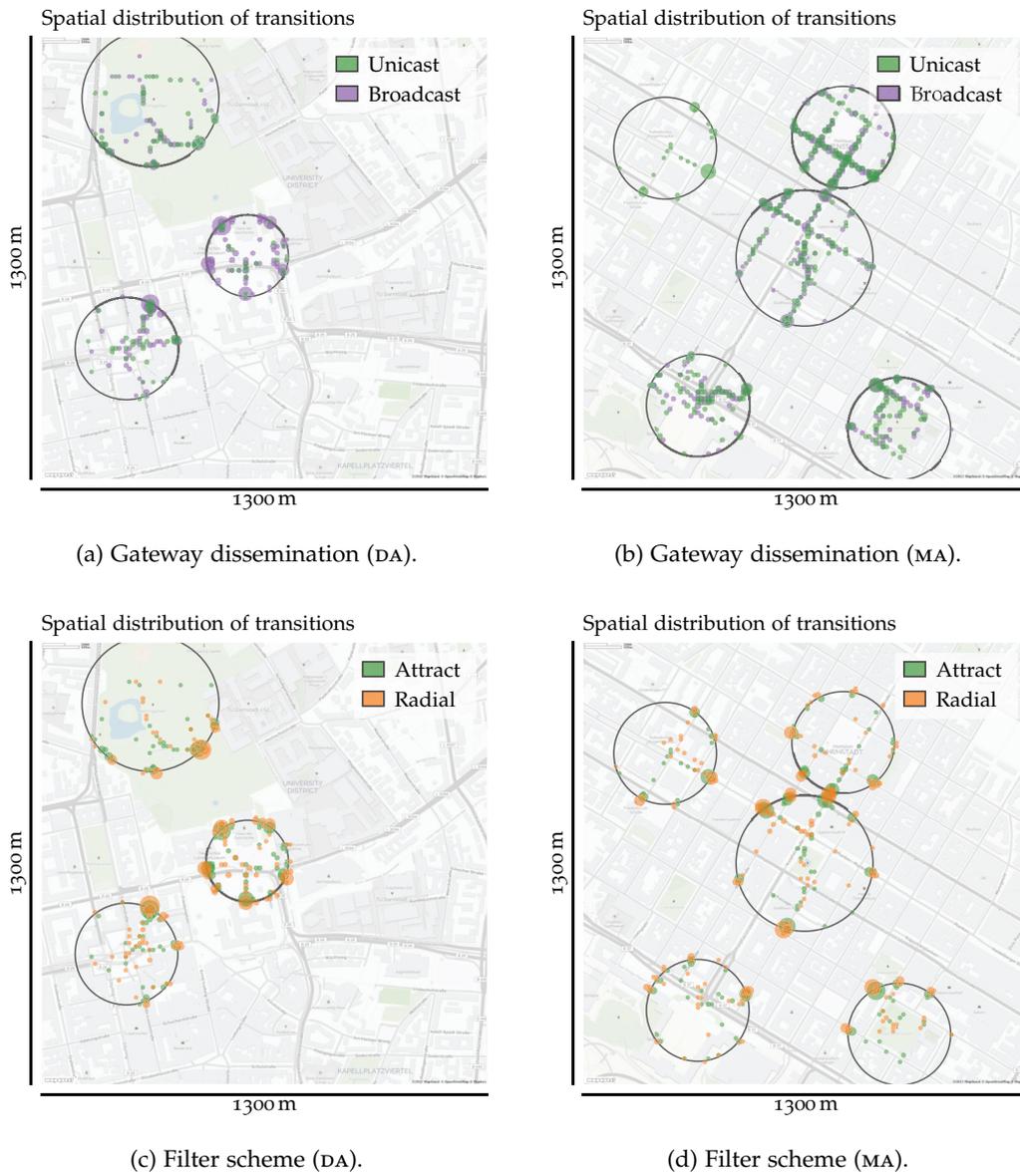


Figure 60: Client locations when executing transitions with $\delta = 8$.

in time all clients use BROADCAST as their gateway dissemination mechanism. Once the client density starts decreasing again (after minute 18), this effect is reversed and clients are again instructed to use UNICAST around most attraction points. The discrete steps visible in the plots correspond to an adaptation of the utilized mechanism for all clients in an attraction point. For $\delta = 10$ one can observe oscillations between both mechanisms. In this case, the client density at an attraction point oscillates around the value of δ , leading to the observed behavior.

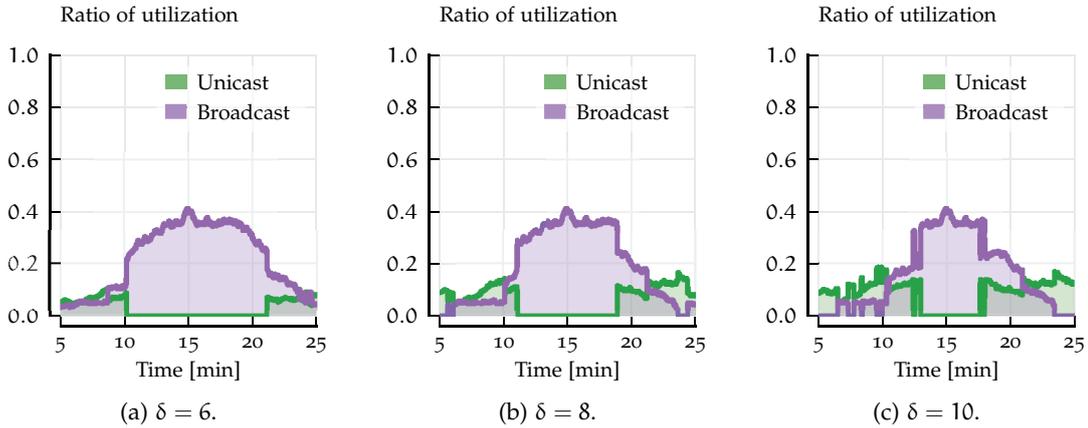


Figure 61: Gateway dissemination mechanism utilization over time (DA).

Notably, these oscillations do not have any visible effect on the performance, as reported in Figure 59. This is due to the fact that oscillations occur at relatively low frequencies compared to the time it takes to distribute an execution plan and execute the respective transition on all affected clients. For reference, a global transition affecting all clients is executed within approximately 50 ms counted from the execution at the first client, as reported in Section 6.3.3. Consequently, the effect of locally mismatching dissemination mechanisms is not of concern in this scenario. Nevertheless, to avoid oscillations and the associated control overhead, one could add additional hysteresis to the respective execution plan introduced in Section 5.3.2.

Oscillations

Individual mechanism utilization per host is reported in Figure 62. With increasing δ , the utilization shifts from BROADCAST to UNICAST. The duration neither BROADCAST nor UNICAST is utilized remains constant regardless of δ , as it is solely determined by the mobility model. As the reported results utilize the same random seed for the mobility model, this behavior does not differ.

We report the results for the MA scenario in Figure 63 for comparison. Due to the larger number of attraction points, the overall behavior over time is more diverse than in the DA scenario. Still, discrete jumps in mechanism utilization over time are visible in all cases. As shown in the spatial distribution of transition reported for $\delta = 8$ in Figure 60b, at some attraction points no transition between dissemination mechanisms is triggered at all. For larger values of δ as reported in Figure 63c, this is the case for an even larger fraction of the attraction points.

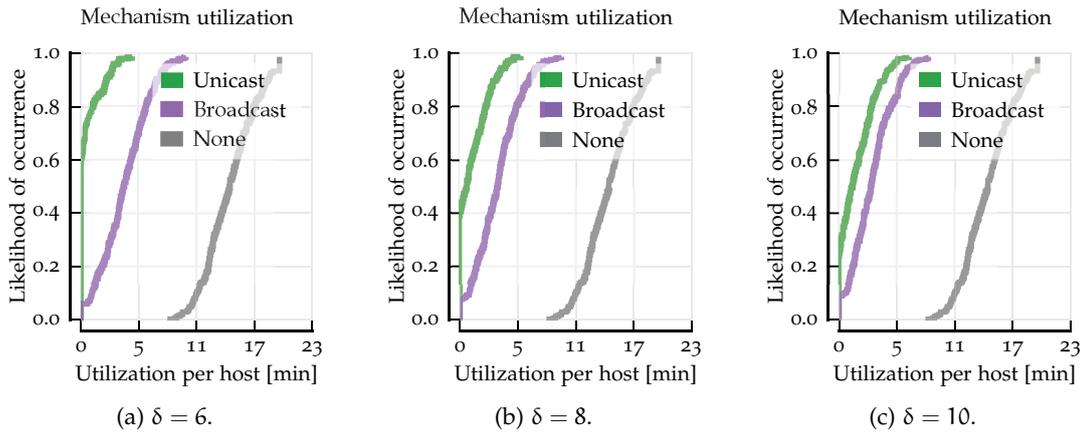


Figure 62: Gateway dissemination mechanism distribution over clients (DA).

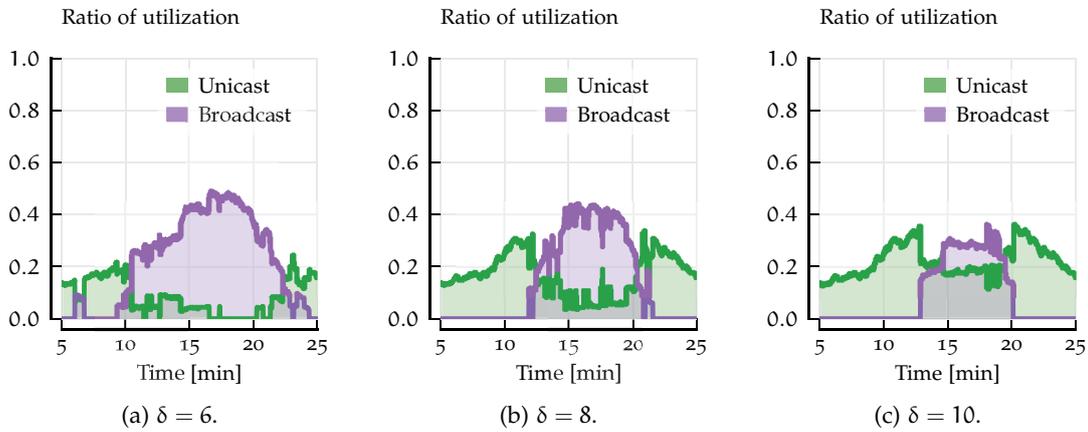


Figure 63: Gateway dissemination mechanism utilization over time (MA).

In the following, we discuss the coordination overhead resulting from the dissemination of transition execution plans to mobile clients. We further show the potential of using the self-healing mechanism as proposed in Section 5.3.2 to reduce the overhead of central coordination. Figure 64a reports the number of messages sent to control transitions for different values of δ , comparing central coordination with self-healing. We are not interested in the absolute message sizes, as it significantly depends on how the transition execution plan is encoded and compressed. This technical aspect is not addressed in our thesis. In the self-heal approach, the execution plans used to adapt the local dissemination mechanisms is not actively disseminated. Clients entering the area of an attraction point are instructed to use the new mechanism. The decision then spreads as a side effect of event dissemination due to our self-healing mechanism proposed in Section 4.3.3.

As expected, the self-heal approach leads to transition coordination traffic that is independent of the chosen value for δ . In contrast, if the decision is to be distributed to all clients that are currently subscribed to an attraction point channel, the in-

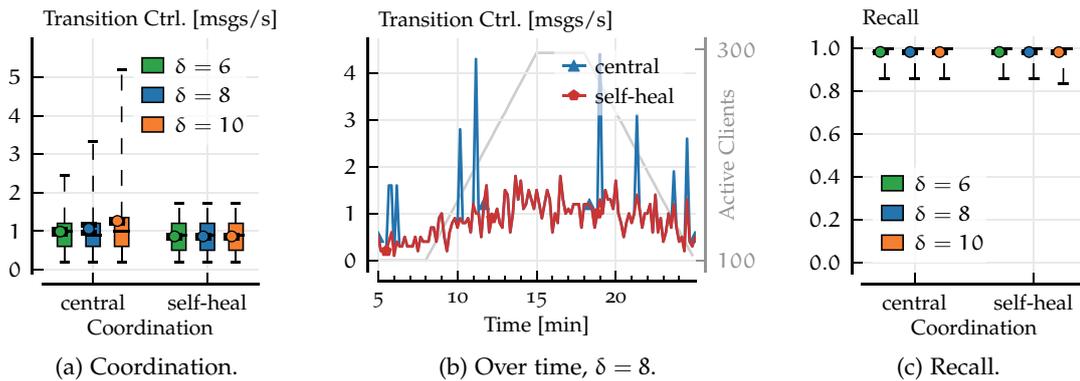


Figure 64: Central coordination of transitions compared to self-healing (DA).

creasing number of transitions for higher δ leads to increased coordination overhead whenever the local dissemination mechanism is to be changed.

This is visible in Figure 64b, where the coordination traffic is reported over time for $\delta = 8$. Both coordination schemes behave identical in terms of triggering transitions for clients that enter or leave the area around an attraction point. However, for central coordination, significant spikes are observed whenever all clients within a channel have to adapt their local dissemination mechanism. Using the self-healing mechanism does not lead to a degradation of the overall performance, as reported in Figure 64c. Only for $\delta = 10$, a slight degradation is visible for the reported 2.5th percentile. This effect can be mitigated if oscillations are avoided.

In this part of our evaluation, we discussed the utilization of mechanism transitions to adapt to application-specific workloads and mobility characteristics. To this end, we evaluated the utilization of direct ad hoc dissemination of local events within vicinity of attraction points in Section 6.4.1. We showed that the dissemination delay is reduced significantly. This enables more interactive applications and can, for example, be utilized to exchange information that reduces the delay of visual effects caused by other users' actions within an augmented reality application. Combined with the utilization of gateways as evaluated in Section 6.4.2, the load on the cellular network can be reduced by up to 64% in our scenario, without any negative effects on the overall performance of the publish/subscribe system. Finally, in the last section, we evaluated the potential of adapting the respective execution plans and, thereby, the choice of dissemination mechanisms, based on the observed client density around attraction points. We further discussed the overhead caused by the coordination of transitions and demonstrated how the self-healing mechanism can be utilized to significantly reduce the cost of this process.

Overall, we successfully demonstrated the combined and adaptive utilization of mechanisms for location-based filtering and locality-aware dissemination of events through the execution of mechanism transitions.

SUMMARY, CONCLUSIONS, AND OUTLOOK

To conclude our work, we summarize the content of the previous chapters and state our main contributions in the following. We then draw conclusions based on our obtained results. Finally, we discuss open issues and potential future work.

7.1 SUMMARY OF THE THESIS

In Chapter 1, we described the challenges for a communication system that result from location-based mobile social applications. We motivated the utilization of publish/subscribe as communication paradigm to address the interaction among users typical for these applications, described in detail in Chapter 2. To address dynamics arising from user mobility and application-specific attraction points, we motivated the design concept of mechanism transitions. We studied existing mechanisms for location-based filtering and locality-aware dissemination and proposed their combined utilization in Chapter 3. Additionally, we discussed related approaches to adaptivity in publish/subscribe and studied the utilization of mechanism transitions in other application domains. Based on our analysis of the state of the art, we presented the following contributions in our thesis.

7.1.1 Contributions

BYPASS.KOM, introduced in Chapter 4, is a framework to study the potential of mechanism transitions for location-based filtering and locality-aware dissemination of events, constituting our *first contribution*. In this framework, we presented a methodology and, consequently, a design for transitions between filter schemes for location-based publish/subscribe. We proposed an encapsulation method for filter schemes, consisting of client and broker components, a scheme-specific context update protocol, and the respective storage for context information at the broker. Thereupon, we discussed the integration of context-based and channel-based filter schemes and designed transitions to switch between filter schemes of both types at runtime. Our design includes a state transfer mechanism to address the challenge of seamless execution of transitions. In contrast to existing work on location-based publish/subscribe, our design is not limited to parameter adaptations within a filter scheme. Instead, existing and upcoming filter schemes can be utilized by integrating them as transition-enabled mechanisms into BYPASS.KOM.

*Transitions
between filter
schemes*

In addition to location-based filtering, we addressed locality-aware event dissemination by enabling transitions between local dissemination mechanisms in BYPASS.KOM. We extended and applied our methodology for the encapsulation of filter schemes to ad hoc dissemination mechanisms. Thereupon, we proposed hybrid modes of

*Locality-
aware event
dissemination*

operation that specifically address distinct communication interfaces available on today's mobile devices. We explicitly distinguished between local events (being relevant within vicinity of their publisher) and generic location-based events targeted at arbitrary locations. This enabled us to benefit from geofencing and locally confined dissemination mechanisms. We further proposed a self-healing mechanism to resolve situations where mobile clients utilize different dissemination mechanisms as a potential consequence of transitions. Existing approaches to locality-aware event dissemination focus either solely on local ad hoc networks, or are limited to the organization of distributed broker networks. Additionally, they do not consider the peculiarities of location-based publish/subscribe, leading to unnecessary overhead in the dissemination of events. In contrast, our approach of transition-enabled event dissemination mechanisms combined with hybrid modes of operation allowed us to tackle the challenge of efficient utilization of heterogeneous networks.

Finally, we proposed the combined utilization of mechanisms for location-based filtering and locality-aware event dissemination in `BYPASS.COM` depending on application-specific attraction points to address the mobility characteristics of mobile social applications. To coordinate transitions between coexisting mechanisms, we modeled the respective transitions through execution plans. We introduced self-transitions as a generic concept to adapt a mechanism's state or configuration parameters. Thereby, we can utilize two distinct concepts for adaptivity, (parameter) reconfiguration, and mechanism transitions in a mechanism-independent fashion. Current adaptive publish/subscribe middleware either does not offer the ability to completely exchange mechanisms at all, or it offers only global reconfiguration.

*Environment
for the design
and
evaluation of
transitions*

With our *second contribution*, we generalized our findings by contributing transition-specific design abstractions and components to the `SIMONSTRATOR.COM` platform, as discussed in Chapter 5. Further, we contributed mobility and network models for the scenario of mobile social applications. The `SIMONSTRATOR.COM` platform and our transition-specific contributions constitute the foundation for the evaluation and characterization of mechanism transitions presented in Chapter 6.

7.1.2 Conclusions

We conducted an extensive evaluation to assess the effects of transitions between different mechanisms on a publish/subscribe system. We showed that our transition-enabled system enables the utilization of individual mechanisms with their respective performance and cost characteristics. It correctly executes transitions between mechanisms, both for location-based filtering and locality-aware event dissemination. Consequently, by integrating the respective mechanisms into our framework, the overall system can adapt to a broad range of conditions, as envisioned in the Collaborative Research Centre "MAKI". By transferring state during a transition, communication between clients and the broker during initialization of the target mechanism is avoided. We compared the behavior of location-based filter schemes with and without state transfer, showing that the mechanism substantially contributes to a seamless execution of transitions. We further evaluated our concept of partial

*Characteriz-
ing
transitions*

*Seamless
transition
execution*

transitions that switch the filter scheme for a subset of clients, motivated by the need for local adaptation to application-specific workload and mobility characteristics. We showed that a broker can execute a partial transition without requiring any client-side modifications. We argue that partial transitions constitute an important foundation for systems that need to adapt to heterogeneous characteristics.

Partial and total transitions

We showed that pure ad hoc dissemination of events is not feasible for location-based mobile social applications, given that such dissemination mechanisms are unable to cover larger distances. Still, our results motivate the utilization of direct ad hoc dissemination for locally relevant events. We evaluated the combination of location-based filter schemes and locality-aware event dissemination mechanisms, proposing the utilization of gateway selection algorithms to further offload the cellular connection. Our evaluation results show that locally confined partial transitions around application-specific attraction points enable the publish/subscribe system to adapt to heterogeneous requirements, while at the same time maintaining high performance. We show how the self-healing mechanism proposed for locality-aware dissemination schemes can further reduce the coordination cost of transitions.

Combined utilization of mechanisms

Lastly, we showed the applicability of our proposed generic design concepts for transition-enabled communication systems. By applying the abstractions contributed to the SIMONSTRATOR.KOM platform, transitions between publish/subscribe mechanisms are coordinated and executed in a mechanism-independent fashion. Most notably, the proposed concept of self-transitions enables fine-grained reconfiguration of the utilized transition-enabled mechanisms in a unified fashion. We demonstrated the coordinated execution of total transitions, partial transitions, and self-transitions through execution plans. We evaluated the adaptation of the execution plans to environmental parameters such as the currently observed client density, further demonstrating the potential of mechanism transitions as a generic way to realize adaptivity in communication systems.

Applicability of generic concepts

7.2 OUTLOOK

The results presented in this thesis motivate the evaluation of transition-enabled publish/subscribe systems in other application scenarios, where high adaptivity to the respective application-specific characteristics is required. Especially in the automotive sector, location-based filtering and locality-aware dissemination of information is a key requirement for assisted or autonomous driving as well as traditional assistance systems. In contrast to the work presented in this thesis, a distributed coordination of transitions might be required, as cellular connectivity cannot always be assumed. Moreover, locality-aware event dissemination mechanisms could benefit from additional locally available information, e. g., more accurate positioning or on-board sensors for speed and nearby vehicles.

Automotive scenarios

Given the hybrid utilization of cellular and local ad hoc connectivity, the respective dissemination mechanisms could benefit from Software-defined wireless networks (SDWNs). Utilizing a SDWN, native multicasting of content to interested clients nearby could be utilized, for example, for clients associated to a Wi-Fi access point or

Software-defined wireless networks

an LTE cell. Similar to the popularity of SDN in fixed networks, SDWNs are expected to act as an enabler for more efficient dissemination mechanisms. This could include filter capabilities within the network itself, as already proposed for SDN [93, 94].

*Proactive
transition
execution*

As briefly discussed in Chapter 1, we considered the reactive execution of transitions in the scope of this work. However, utilizing knowledge on application-specific behavioral patterns to execute transitions in a proactive fashion could further contribute to the seamlessness of transition execution. Proactive planning and execution of transitions is a core research topic in the second funding period of the Collaborative Research Centre “MAKI”.

*Location
privacy*

Lastly, transitions between filter schemes in a location-based publish/subscribe system can also act as an enabler for increased location privacy [130]. Instead of subscribing to an exact location (and updating that location frequently), a user might decide to switch to a less accurate filter scheme that does not require accurate location information. Such user-initiated mechanism transitions would enable users to select their preferred privacy vs. accuracy trade-off transparent to the application.

Our contributions to the realization and evaluation of mechanism transitions in publish/subscribe systems and their generalization in the SIMONSTRATOR.KOM platform constitute the foundation for further research in the aforementioned directions.

ACKNOWLEDGMENTS

This work has been funded as part of projects C2 and C3 by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 “MAKI – Multi-Mechanisms Adaptation for the Future Internet”.

BIBLIOGRAPHY

- [1] Osama Abboud. "Quality adaptation in peer-to-peer video streaming: Supporting heterogeneity and enhancing performance using scalable video coding." PhD thesis. TU Darmstadt, 2012.
- [2] VS Abhayawardhana, IJ Wassell, D Crosby, MP Sellars, and MG Brown. "Comparison of empirical propagation path loss models for fixed wireless access systems." In: *Proc. IEEE Vehicular Technology Conference (VTC)*. Vol. 1. IEEE. 2005, pp. 73–77.
- [3] Tim Althoff, Ryen W White, and Eric Horvitz. "Influence of Pokémon Go on Physical Activity: Study and Implications." In: *arXiv preprint arXiv:1610.02085* (2016).
- [4] Ken Arnold, Robert Scheifler, Jim Waldo, Bryan O'Sullivan, and Ann Wollrath. *Jini specification*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [5] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc Networks." In: *Proc. IEEE International Conference on Computer Communications (INFOCOM)*. Vol. 2. IEEE. 2003, pp. 825–835.
- [6] Roberto Baldoni, Carlo Marchetti, Antonino Virgillito, and Roman Vitenberg. "Content-based publish-subscribe over structured overlay networks." In: *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2005, pp. 437–446.
- [7] Paul Baumann, Johannes Klaus, Björn Richerzhagen, Wilhelm Kleiminger, and Silvia Santini. "The UbiDCC Data Set: Collecting Wi-Fi and Bluetooth Scans During a Large-scale Conference." In: *Proc. 1st Workshop on Context Sensing and Activity Recognition (CSAR)*. ACM. 2015, pp. 25–30.
- [8] Paolo Bellavista, Antonio Corradi, Mario Fanelli, and Luca Foschini. "A survey of context data distribution for mobile ubiquitous systems." In: *ACM Computing Surveys (CSUR)* 44.4 (2012), p. 24.
- [9] Mark Berman, Piet Demeester, Jae Woo Lee, Kiran Nagaraja, Michael Zink, Didier Colle, Dilip Kumar Krishnappa, Dipankar Raychaudhuri, Henning Schulzrinne, Ivan Seskar, et al. "Future internets escape the simulator." In: *Communications of the ACM* 58.6 (2015), pp. 78–89.
- [10] Giuseppe Bianchi. "Performance analysis of the IEEE 802.11 distributed coordination function." In: *IEEE Journal on selected areas in communications* 18.3 (2000), pp. 535–547.
- [11] Allan Brimicombe and Yang Li. "Mobile Space-Time Envelopes for Location-Based Services." In: *Transactions in GIS* 10.1 (2006), pp. 5–23.

- [12] Andrei Broder and Michael Mitzenmacher. "Network applications of bloom filters: A survey." In: *Internet mathematics* 1.4 (2004), pp. 485–509.
- [13] Ioana Burcea and Hans-Arno Jacobsen. "L-ToPSS—push-oriented location-based services." In: *International Workshop on Technologies for E-Services*. Springer. 2003, pp. 131–142.
- [14] Claudia Campolo, Antonella Molinaro, and Riccardo Scopigno. *Vehicular ad hoc Networks: Standards, Solutions, and Research*. Springer, 2015.
- [15] Mauro Caporuscio, Antonio Carzaniga, and Alexander L Wolf. "Design and evaluation of a support service for mobile, wireless publish/subscribe applications." In: *IEEE Transactions on Software Engineering* 29.12 (2003), pp. 1059–1071.
- [16] Antonio Carzaniga, David S Rosenblum, and Alexander L Wolf. "Achieving scalability and expressiveness in an internet-scale event notification service." In: *Proc. ACM Symposium on Principles of Distributed Computing*. ACM. 2000, pp. 219–227.
- [17] Antonio Carzaniga, David S Rosenblum, and Alexander L Wolf. "Design and evaluation of a wide-area event notification service." In: *ACM Transactions on Computer Systems (TOCS)* 19.3 (2001), pp. 332–383.
- [18] Mainak Chatterjee, Sajal K Das, and Damla Turgut. "WCA: A weighted clustering algorithm for mobile ad hoc networks." In: *Cluster computing* 5.2 (2002), pp. 193–204.
- [19] Lisi Chen, Gao Cong, Xin Cao, and Kian-Lee Tan. "Temporal spatial-keyword top-k publish/subscribe." In: *Proc. International Conference on Data Engineering*. IEEE. 2015, pp. 255–266.
- [20] Xiaoyan Chen, Ying Chen, and Fangyan Rao. "An efficient spatial publish/subscribe system for intelligent location-based services." In: *Proc. International Workshop on Distributed Event-based Systems (DEBS)*. ACM. 2003, pp. 1–6.
- [21] Ying Chen, Fangyan Rao, Xiulan Yu, and Dong Liu. "CAMEL: A moving object database approach for intelligent location aware services." In: *Proc. IEEE International Conference on Mobile Data Management (MDM)*. Springer. 2003, pp. 331–334.
- [22] Alex King Yeung Cheung and Hans-Arno Jacobsen. "Publisher placement algorithms in content-based publish/subscribe." In: *Proc. International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2010, pp. 653–664.
- [23] Imrich Chlamtac, Marco Conti, and Jennifer J-N Liu. "Mobile ad hoc networking: imperatives and challenges." In: *Ad hoc networks* 1.1 (2003), pp. 13–64.
- [24] Mariano Cilia, Ludger Fiege, Christian Haul, Andreas Zeidler, and Alejandro P Buchmann. "Looking into the past: enhancing mobile publish/subscribe middleware." In: *Proc. International Workshop on Distributed Event-based Systems (DEBS)*. ACM. 2003, pp. 1–8.

- [25] Cisco. *Cisco Visual Networking Index: Forecast and Methodology, 2015–2020*. June 2016. URL: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html> (Last accessed on March 8, 2017).
- [26] Cisco. *The Zettabyte Era: Trends and Analysis*. July 2016. URL: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html> (Last accessed on March 8, 2017).
- [27] Marco Conti and Silvia Giordano. “Mobile ad hoc networking: milestones, challenges, and new research directions.” In: *IEEE Communications Magazine* 52.1 (2014), pp. 85–96.
- [28] Paolo Costa, Cecilia Mascolo, Mirco Musolesi, and Gian Pietro Picco. “Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks.” In: *IEEE Journal on Selected Areas in Communications* 26.5 (2008), pp. 748–760.
- [29] Paolo Costa and Gian Pietro Picco. “Semi-probabilistic content-based publish-subscribe.” In: *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2005, pp. 575–585.
- [30] Gianpaolo Cugola, Alessandro Margara, and Matteo Migliavacca. “Context-aware publish-subscribe: Model, implementation, and evaluation.” In: *Proc. IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2009, pp. 875–881.
- [31] Gianpaolo Cugola and Gian Pietro Picco. “REDS: a reconfigurable dispatching system.” In: *Proc. International Workshop on Software Engineering and Middleware (SEM)*. ACM. 2006, pp. 9–16.
- [32] Anwitaman Datta, Silvia Quarteroni, and Karl Aberer. “Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks.” In: *Semantics of a Networked World. Semantics for Grid Databases*. Springer, 2004, pp. 126–143.
- [33] Alan Demers, Johannes Gehrke, Mingsheng Hong, Mirek Riedewald, and Walker White. “Towards expressive publish/subscribe systems.” In: *International Conference on Extending Database Technology*. Springer. 2006, pp. 627–644.
- [34] Subhankar Dhar and Upkar Varshney. “Challenges and Business Models for Mobile Location-based Services and Advertising.” In: *Communications of the ACM* 54.5 (2011), pp. 121–128.
- [35] Shuo Ding. “A survey on integrating MANETs with the Internet: Challenges and designs.” In: *Computer Communications* 31.14 (2008), pp. 3537–3551.
- [36] Danny Dolev, Cynthia Dwork, and Larry Stockmeyer. “On the minimal synchronism needed for distributed consensus.” In: *Journal of the ACM (JACM)* 34.1 (1987), pp. 77–97.

- [37] Dorine C Duives, Winnie Daamen, and Serge P Hoogendoorn. "State-of-the-art crowd motion simulation models." In: *Transportation research part C: emerging technologies* 37 (2013), pp. 193–209.
- [38] Tim Dutz. "Pervasive Behavior Interventions-Using Mobile Devices for Overcoming Barriers for Physical Activity." PhD thesis. Technische Universität Darmstadt, 2017.
- [39] Ericsson. *Ericsson Mobility Report: on the Pulse of the Networked Society*. June 2016. URL: <https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf> (Last accessed on March 8, 2017).
- [40] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Ker-marrec. "The Many Faces of Publish/Subscribe." In: *ACM Computing Surveys (CSUR)* 35.2 (2003), pp. 114–131.
- [41] Patrick Th Eugster, Benott Garbinato, and Adrian Holzer. "Location-based publish/subscribe." In: *Proc. IEEE International Symposium on Network Computing and Applications*. IEEE. 2005, pp. 279–282.
- [42] Patrick Eugster and KR Jayaram. "EventJava: An extension of Java for event correlation." In: *European Conference on Object-Oriented Programming*. Springer. 2009, pp. 570–594.
- [43] Ludger Fiege, Felix C Gärtner, Oliver Kasten, and Andreas Zeidler. "Supporting mobility in content-based publish/subscribe middleware." In: *Proc. ACM/IFIP/USENIX 2003 International Conference on Middleware*. Springer-Verlag New York, Inc. 2003, pp. 103–122.
- [44] Davide Frey and Gruia-Catalin Roman. "Context-aware publish subscribe in mobile ad hoc networks." In: *Proc. International Conference on Coordination Languages and Models*. Springer. 2007, pp. 37–55.
- [45] Roy Friedman and Anna Kaplun Shulman. "A density-driven publish subscribe service for mobile ad-hoc networks." In: *Ad Hoc Networks* 11.1 (2013), pp. 522–540.
- [46] Alexander Frömmgen, Mohamed Hassan, Roland Kluge, Mahdi Mousavi, Max Mühlhäuser, Sabrina Müller, Mathias Schnee, Michael Stein, and Markus Weckesser. *Mechanism Transitions: A New Paradigm for a Highly Adaptive Internet*. Tech. rep. Darmstadt, Mar. 2016. URL: <http://tuprints.ulb.tu-darmstadt.de/5370/>.
- [47] Alexander Frömmgen, Robert Rehner, Max Lehn, and Alejandro Buchmann. "Fossa: Learning eca rules for adaptive distributed systems." In: *Proc. IEEE International Conference on Autonomic Computing (ICAC)*. IEEE. 2015, pp. 207–210.
- [48] Alexander Frömmgen, Robert Rehner, Max Lehn, and Alejandro Buchmann. "Fossa: Using genetic programming to learn ECA rules for adaptive networking applications." In: *Proc. IEEE Conference on Local Computer Networks (LCN)*. IEEE. 2015, pp. 197–200.

- [49] Alexander Frömmgen, Björn Richerzhagen, Julius Rückert, David Hausheer, Ralf Steinmetz, and Alejandro Buchmann. "Towards the Description and Execution of Transitions in Networked Systems." In: *Proc. 9th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*. IFIP. June 2015, pp. 17–29.
- [50] Zhenghua Fu, Xiaoqiao Meng, and Songwu Lu. "How bad TCP can perform in mobile ad hoc networks." In: *Proc. International Symposium on Computers and Communications (ISCC)*. IEEE. 2002, pp. 298–303.
- [51] Abdulbaset Gaddah and Thomas Kunz. "Extending mobility to publish/subscribe systems using a pro-active caching approach." In: *Mobile Information Systems 6.4 (2010)*, pp. 293–324.
- [52] Huiji Gao, Jiliang Tang, and Huan Liu. "Exploring Social-Historical Ties on Location-Based Social Networks." In: *ICWSM*. 2012.
- [53] Sarunas Girdzijauskas, Gregory Chockler, Ymir Vigfusson, Yoav Tock, and Roie Melamed. "Magnet: practical subscription clustering for internet-scale publish/subscribe." In: *Proc. ACM International Conference on Distributed Event-Based Systems (DEBS)*. ACM. 2010, pp. 172–183.
- [54] Aniruddha Gokhale, Douglas C. Schmidt, and Joe Hoffert. "Timely Automatic Adaptation of Publish/Subscribe Middleware in Dynamic Environments." In: *Int. J. Adapt. Resilient Auton. Syst.* 2.4 (Oct. 2011), pp. 1–24. URL: <http://dx.doi.org/10.4018/jaras.2011100101>.
- [55] Google Inc. *Google Project Fi – Network Coverage*. 2016. URL: <https://fi.google.com/about/coverage/#network-of-networks>.
- [56] Christian Groß, Björn Richerzhagen, and Max Lehn. "Structured Search Overlays." In: *Benchmarking Peer-to-Peer Systems*. Springer Berlin Heidelberg, 2013, pp. 49–67.
- [57] Christian Groß, Björn Richerzhagen, Dominik Stingl, Christoph Munker, David Hausheer, and Ralf Steinmetz. "Geodemlia: Persistent Storage and Reliable Search for Peer-to-Peer Location-based Services." In: *Proc. 13th IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE. 2013, pp. 1–2.
- [58] Christian Groß, Björn Richerzhagen, Dominik Stingl, Jens Weber, David Hausheer, and Ralf Steinmetz. "Geoswarm: A Multi-Source Download Scheme for Peer-to-Peer Location-based Services." In: *Proc. 13th IEEE International Conference on Peer-to-Peer Computing*. IEEE. 2013, pp. 1–10.
- [59] Christian Groß, Dominik Stingl, Christian Gottron, Björn Richerzhagen, Christoph Munker, and David Hausheer. *Harnessing Mobile Ad Hoc Communication for Decentralized Location-Based Services*. Tech. rep. Peer-to-Peer Systems Engineering Lab, TU Darmstadt, Germany, 2012.
- [60] Christian Groß, Dominik Stingl, Björn Richerzhagen, Andreas Hemel, Ralf Steinmetz, and David Hausheer. "Geodemlia: A Robust Peer-to-Peer Overlay Supporting Location-based Search." In: *Proc. 12th IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE. 2012, pp. 25–36.

- [61] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*. Vol. 14. 2. ACM, 1984.
- [62] Rhaban Hark, Nils Richerzhagen, Björn Richerzhagen, Amr Rizk, and Ralf Steinmetz. "Towards an Adaptive Selection of Loss Estimation Techniques in Software-defined Networks." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IEEE. 2017, pp. 1–9.
- [63] Samiul Hasan, Xianyuan Zhan, and Satish V Ukkusuri. "Understanding urban human activity and mobility patterns using large-scale location-based data from online social media." In: *Proc. ACM SIGKDD International Workshop on Urban Computing*. ACM. 2013, p. 6.
- [64] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks." In: *Proc. International Conference on System Sciences*. IEEE. 2000, 10–pp.
- [65] Jens Heuschkel, Michael Stein, Lin Wang, and Max Mühlhäuser. "Beyond the core: Enabling software-defined control at the network edge." In: *Proc. International Conference on Networked Systems (NetSys)*. IEEE. 2017, pp. 1–6.
- [66] Adrian Holzer, Patrick Eugster, and Benoit Garbinato. "Evaluating implementation strategies for location-based multicast addressing." In: *IEEE Transactions on Mobile Computing* 12.5 (2013), pp. 855–867.
- [67] Adrian Holzer, Patrick Eugster, and Benot Garbinato. "Alps–adaptive location-based publish/subscribe." In: *Computer Networks* 56.12 (2012), pp. 2949–2962.
- [68] Adrian Holzer, François Vessaz, Samuel Pierre, and Benoit Garbinato. "PLAN-B: Proximity-based lightweight adaptive network broadcasting." In: *Proc. International Symposium on Network Computing and Applications (NCA)*. IEEE. 2011, pp. 265–270.
- [69] Adrian Holzer, Lukasz Ziarek, KR Jayaram, and Patrick Eugster. "Abstracting context in event-based software." In: *Transactions on Aspect-Oriented Software Development IX*. Springer, 2012, pp. 123–167.
- [70] Songlin Hu, Vinod Muthusamy, Guoli Li, and Hans-Arno Jacobsen. "Transactional mobility in distributed content-based publish/subscribe systems." In: *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2009, pp. 101–110.
- [71] Yongqiang Huang and Hector Garcia-Molina. "Publish/subscribe in a mobile environment." In: *Proc. ACM international workshop on Data engineering for wireless and mobile access*. ACM. 2001, pp. 27–34.
- [72] Yongqiang Huang and Hector Garcia-Molina. "Publish/subscribe tree construction in wireless ad-hoc networks." In: *Proc. International Conference on Mobile Data Management*. Springer. 2003, pp. 122–140.
- [73] Yongqiang Huang and Hector Garcia-Molina. "Parameterized subscriptions in publish/subscribe systems." In: *Data & Knowledge Engineering* 60.3 (2007), pp. 435–450.

- [74] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. "MQTT-S A publish/subscribe protocol for Wireless Sensor Networks." In: *Proc. International Conference on Communication systems software and middleware and workshops (comsware)*. IEEE. 2008, pp. 791–798.
- [75] Sergio Ilarri, Eduardo Mena, and Arantza Illarramendi. "Location-dependent Query Processing: Where we are and where we are Heading." In: *ACM Computing Surveys (CSUR)* 42.3 (2010), p. 12.
- [76] Raj Jain, Dah-Ming Chiu, and William R Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Vol. 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.
- [77] Kasthuri Jayarajah, Youngki Lee, Archan Misra, and Rajesh Krishna Balan. "Need accurate user behaviour?: pay attention to groups!" In: *Proc. ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 855–866.
- [78] KR Jayaram, Patrick Eugster, and Chamikara Jayalath. "Parametric content-based publish/subscribe." In: *ACM Transactions on Computer Systems (TOCS)* 31.2 (2013), p. 4.
- [79] KR Jayaram, Weihang Wang, and Patrick Eugster. "Subscription normalization for effective content-based messaging." In: *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 26.11 (2015), pp. 3184–3193.
- [80] Gordian Jodlauk, Rene Rembarz, and Zhiwei Xu. "An optimized grid-based geocasting method for cellular mobile networks." In: *Proc. ITS World Congress*. 2011.
- [81] David B Johnson and David A Maltz. "Dynamic source routing in ad hoc wireless networks." In: *Mobile computing*. Springer, 1996, pp. 153–181.
- [82] Fabian Kaup, Florian Fischer, and David Hausheer. "Measuring and predicting cellular network quality on trains." In: *Proc. International Conference on Networked Systems (NetSys)*. IEEE. 2017, pp. 1–8.
- [83] Fabian Kaup, Florian Jomrich, and David Hausheer. "Demonstration of Network Coverage – A Mobile Network Performance Measurement App." In: *Proc. International Conference on Networked Systems (NetSys)*. IEEE. 2015.
- [84] Nipendra Kayastha, Dusit Niyato, Ping Wang, and Ekram Hossain. "Applications, Architectures, and Protocol Design issues for Mobile Social Networks: A Survey." In: *Proceedings of the IEEE* 99.12 (2011), pp. 2130–2158.
- [85] Jeffrey O Kephart and David M Chess. "The vision of autonomic computing." In: *Computer* 36.1 (2003), pp. 41–50.
- [86] Abdelmajid Khelil, Pedro José Marrón, Christian Becker, and Kurt Rothermel. "Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks." In: *Ad Hoc Networks* 5.5 (2007), pp. 531–546.

- [87] Samamon Khemmarat, Renjie Zhou, Dilip Kumar Krishnappa, Lixin Gao, and Michael Zink. "Watching user generated videos with prefetching." In: *Signal Processing: Image Communication* 27.4 (2012), pp. 343–359.
- [88] Roland Kluge, Michael Stein, Gergely Varró, Andy Schürr, Matthias Hollick, and Max Mühlhäuser. "A systematic approach to constructing families of incremental topology control algorithms using graph transformation." In: *Software & Systems Modeling* (2017), pp. 1–41.
- [89] Michael Knappmeyer, Saad Liaquat Kiani, Eike Steffen Reetz, Nigel Baker, and Ralf Tonjes. "Survey of context provisioning middleware." In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1492–1519.
- [90] Richard Knoblauch, Martin Pietrucha, and Marsha Nitzburg. "Field studies of pedestrian walking speed and start-up time." In: *Transportation Research Record: Journal of the Transportation Research Board* 1538 (1996), pp. 27–38.
- [91] Donald Ervin Knuth. *The art of computer programming: sorting and searching*. Vol. 3. Pearson Education, 1998.
- [92] Y-B Ko and Nitin H Vaidya. "Geocasting in mobile ad hoc networks: Location-based multicast algorithms." In: *Proc. IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*. IEEE. 1999, pp. 101–110.
- [93] Boris Koldehofe, Frank Dürr, and Muhammad Adnan Tariq. "Event-based systems meet software-defined networking." In: *Proc. ACM International Conference on Distributed Event-based Systems (DEBS)*. ACM. 2013, pp. 271–280.
- [94] Boris Koldehofe, Frank Dürr, Muhammad Adnan Tariq, and Kurt Rothermel. "The power of software-defined networking: line-rate content-based routing using OpenFlow." In: *Proc. Workshop on Middleware for Next Generation Internet Computing*. ACM. 2012, p. 3.
- [95] Boris Koldehofe, Beate Ottenwälder, Kurt Rothermel, and Umakishore Ramachandran. "Moving range queries in distributed complex event processing." In: *Proc. ACM International Conference on Distributed Event-Based Systems (DEBS)*. ACM. 2012, pp. 201–212.
- [96] Aleksandra Kovacevic. "Peer-To-Peer Location-Based Search: Engineering a Novel Peer-To-Peer Overlay Network." PhD thesis. TU Darmstadt, 2009.
- [97] Axel Küpper. *Location-based Services: Fundamentals and Operation*. John Wiley & Sons, 2005.
- [98] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. "MANET simulation studies: the incredibles." In: *ACM SIGMOBILE Mobile Computing and Communications Review* 9.4 (2005), pp. 50–61.
- [99] Stuart Kurkowski, William Navidi, and Tracy Camp. "Constructing manet simulation scenarios that meet standards." In: *Proc. IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*. IEEE. 2007, pp. 1–9.
- [100] Leslie Lamport. "Time, clocks, and the ordering of events in a distributed system." In: *Communications of the ACM* 21.7 (1978), pp. 558–565.

- [101] Max Lehn. "InterestCast: Adaptive Event Dissemination for Interactive Real-Time Applications." PhD thesis. Technische Universität Darmstadt, 2016.
- [102] Max Lehn, Robert Rehner, and Alejandro Buchmann. "Distributed optimization of event dissemination exploiting interest clustering." In: *Proc. IEEE Conference on Local Computer Networks (LCN)*. IEEE. 2013, pp. 328–331.
- [103] Ilias Leontiadis, Paolo Costa, and Cecilia Mascolo. "A hybrid approach for content-based publish/subscribe in vehicular networks." In: *Pervasive and Mobile Computing* 5.6 (2009), pp. 697–713.
- [104] Guoli Li, Shuang Hou, and Hans-Arno Jacobsen. "A unified approach to routing, covering and merging in publish/subscribe systems based on modified binary decision diagrams." In: *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2005, pp. 447–457.
- [105] Guoli Li, Vinod Muthusamy, and Hans-Arno Jacobsen. "Adaptive content-based routing in general overlay topologies." In: *Proc. ACM/IFIP/USENIX International Conference on Middleware*. Springer-Verlag New York, Inc. 2008, pp. 1–21.
- [106] Guoliang Li, Yang Wang, Ting Wang, and Jianhua Feng. "Location-aware publish/subscribe." In: *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2013, pp. 802–810.
- [107] Jinyang Li, Charles Blake, Douglas SJ De Couto, Hu Imm Lee, and Robert Morris. "Capacity of ad hoc wireless networks." In: *Proc. International Conference on Mobile Computing and Networking*. ACM. 2001, pp. 61–69.
- [108] Steve Liang, Yang Gao, et al. "Real-time notification and improved situational awareness in fire emergencies using geospatial-based publish/subscribe." In: *International Journal of Applied Earth Observation and Geoinformation* 12.6 (2010), pp. 431–438.
- [109] Chunhung Richard Lin and Mario Gerla. "Adaptive clustering for mobile wireless networks." In: *IEEE Journal on Selected areas in Communications* 15.7 (1997), pp. 1265–1275.
- [110] Elvis S Liu and Georgios K Theodoropoulos. "Interest management for distributed virtual environments: A survey." In: *ACM Computing Surveys (CSUR)* 46.4 (2014), p. 51.
- [111] George R MacCartney, Junhong Zhang, Shuai Nie, and Theodore S Rappaport. "Path loss models for 5G millimeter wave propagation channels in urban microcells." In: *Proc. IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2013, pp. 3948–3953.
- [112] Christian Maihofer. "A survey of geocast routing protocols." In: *IEEE Communications Surveys & Tutorials* 6.2 (2004).
- [113] Derek Hylton Maling. *Coordinate systems and map projections*. Elsevier, 2013.

- [114] Martin Mauve, Jorg Widmer, and Hannes Hartenstein. "A survey on position-based routing in mobile ad hoc networks." In: *IEEE network* 15.6 (2001), pp. 30–39.
- [115] Rene Meier and Vinny Cahill. "On event-based middleware for location-aware mobile applications." In: *IEEE Transactions on Software Engineering* 36.3 (2010), pp. 409–430.
- [116] René Meier and Vinny Cahill. "Steam: Event-based middleware for wireless ad hoc networks." In: *Proc. International Conference on Distributed Computing Systems Workshops*. IEEE. 2002, pp. 639–644.
- [117] Foivos Michelinakis, Nicola Bui, Guido Fioravanti, Joerg Widmer, Fabian Kaup, and David Hausheer. "Lightweight mobile bandwidth availability measurement." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IEEE. 2015, pp. 1–9.
- [118] Foivos Michelinakis, Nicola Bui, Guido Fioravanti, Joerg Widmer, Fabian Kaup, and David Hausheer. "Lightweight capacity measurements for mobile networks." In: *Computer Communications* 84 (2016), pp. 73–83.
- [119] Pramita Mitra and Christian Poellabauer. "Efficient group communications in location aware mobile ad-hoc networks." In: *Pervasive and Mobile Computing* 8.2 (2012), pp. 229–248.
- [120] Sang-chul Moon, Yangwoo Ko, and Dongman Lee. "A fast path recovery scheme for publish/subscribe in mobile ad hoc networks." In: *Proc. IEEE Conference on Computer and Information Technology (CIT)*. IEEE. 2007, pp. 435–440.
- [121] Luca Mottola, Gianpaolo Cugola, and Gian Pietro Picco. "A self-repairing tree topology enabling content-based routing in mobile ad hoc networks." In: *IEEE Transactions on Mobile Computing* 7.8 (2008), pp. 946–960.
- [122] Mahdi Mousavi, Sabrina Müller, Hussein Al-Shatri, Bernd Freisleben, and Anja Klein. "Multi-hop data dissemination with selfish nodes: Optimal decision and fair cost allocation based on the Shapley value." In: *Proc. IEEE International Conference on Communications (ICC)*. IEEE. 2016, pp. 1–6.
- [123] Mahdi Mousavi, Hussein Al-Shatri, Oliver Hinz, and Anja Klein. "Incorporating User Willingness for Message Forwarding in Multi-Hop Content Distribution Scenarios." In: *Proc. International ITG Workshop on Smart Antennas (WSA)*. VDE. 2016, pp. 1–6.
- [124] Gero Mühl, Ludger Fiege, and Alejandro Buchmann. "Filter similarities in content-based publish/subscribe systems." In: *Trends in Network and Pervasive Computing – ARCS 2002*. Springer, 2002, pp. 224–238.
- [125] Gero Muhl, Ludger Fiege, Felix C Gartner, and Alejandro Buchmann. "Evaluating advanced routing algorithms for content-based publish/subscribe systems." In: *Proc. IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS)*. IEEE. 2002, pp. 167–176.

- [126] Gero Muhl, Andreas Ulbrich, and K Herrman. "Disseminating information to mobile clients using publish-subscribe." In: *IEEE Internet Computing* 8.3 (2004), pp. 46–53.
- [127] Mahdi Tayarani Najaran, Shun-Yun Hu, and Norman C Hutchinson. "SPEX: scalable spatial publish/subscribe for distributed virtual worlds without borders." In: *Proc. ACM Multimedia Systems Conference (MMSys)*. ACM. 2014, pp. 127–138.
- [128] Julio C Navas and Tomasz Imielinski. "GeoCast – geographic addressing and routing." In: *Proc. ACM/IEEE International Conference on Mobile Computing and Networking*. ACM. 1997, pp. 66–76.
- [129] Ben Newton, Jay Aikat, and Kevin Jeffay. "Geographic Routing in Extreme-Scale Highly-Dynamic Mobile Ad Hoc Networks." In: *Proc. IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE. 2016, pp. 205–210.
- [130] Emanuel Onica, Pascal Felber, Hugues Mercier, and Etienne Rivière. "Confidentiality-Preserving Publish/Subscribe: A Survey." In: *ACM Computing Surveys (CSUR)* 49.2 (2016), p. 27.
- [131] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, Kirak Hong, David Lil- lethun, and Umakishore Ramachandran. "Mcep: A mobility-aware complex event processing system." In: *ACM Transactions on Internet Technology (TOIT)* 14.1 (2014), p. 6.
- [132] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, and Umakishore Ra- machandran. "Migcep: Operator migration for mobility driven distributed complex event processing." In: *Proc. ACM International Conference on Distributed Event-based Systems (DEBS)*. ACM. 2013, pp. 183–194.
- [133] Jay A Patel, Étienne Rivière, Indranil Gupta, and Anne-Marie Kermarrec. "Rappel: Exploiting interest and network locality to improve fairness in publish- subscribe systems." In: *Computer Networks* 53.13 (2009), pp. 2304–2320.
- [134] Larry L Peterson and Bruce S Davie. *Computer networks: a systems approach*. Elsevier, 2007.
- [135] Peter R Pietzuch and Jean M Bacon. "Hermes: A distributed event-based mid- dleware architecture." In: *Proc. Conference on Distributed Computing Systems Workshops*. IEEE. 2002, pp. 611–618.
- [136] Peter Pietzuch, David Eysers, Samuel Kounev, and Brian Shand. "Towards a common api for publish/subscribe." In: *Proc. ACM International Conference on Distributed Event-based Systems (DEBS)*. ACM. 2007, pp. 152–157.
- [137] Konstantin Pussep. "Peer-assisted Video-on-Demand: Cost Reduction and Performance Enhancement for Users, Overlay Providers, and Network Op- erators." PhD thesis. TU Darmstadt, 2011.

- [138] Filippo Rebecchi, Marcelo Dias De Amorim, Vania Conan, Andrea Passarella, Raffaele Bruno, and Marco Conti. "Data offloading techniques in cellular networks: a survey." In: *IEEE Communications Surveys & Tutorials* 17.2 (2015), pp. 580–603.
- [139] Christoph Rensing, Stephan Tittel, and Ralf Steinmetz. "Location-Based services for technology enhanced learning and teaching." In: *Software Service and Application Engineering*. Springer, 2012, pp. 165–179.
- [140] Björn Richerzhagen. "Supporting Transitions in Peer-to-Peer Video Streaming." In: *Masters thesis, Technische Universität Darmstadt*. 2012.
- [141] Björn Richerzhagen, Boris Koldehofe, and Ralf Steinmetz. "Immense Dynamism." In: *German Research* 37.2 (2015), pp. 24–27.
- [142] Björn Richerzhagen, Nils Richerzhagen, Sophie Schönherr, Rhaban Hark, and Ralf Steinmetz. "Stateless Gateways – Reducing Cellular Traffic for Event Distribution in Mobile Social Applications." In: *Proc. 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. Aug. 2016, pp. 1–9.
- [143] Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe, and Ralf Steinmetz. "Demo: Seamless Transitions Between Filter Schemes for Location-based Mobile Applications." In: *Demonstrations of the 41st IEEE Conference on Local Computer Networks (LCN-Demos)*. Nov. 2016, pp. 1–3.
- [144] Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe, and Ralf Steinmetz. "Seamless Transitions Between Filter Schemes for Location-based Mobile Applications." In: *Proc. 41st IEEE Conference on Local Computer Networks (LCN)*. Nov. 2016, pp. 1–9.
- [145] Björn Richerzhagen, Marc Schiller, Max Lehn, Denis Lapiner, and Ralf Steinmetz. "Transition-enabled Event Dissemination for Pervasive Mobile Multiplayer Games." In: *Proc. 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2015.
- [146] Björn Richerzhagen and Ralf Steinmetz. "Towards an Adaptive Publish/Subscribe Approach Supporting Transitions." In: *Proc. 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS), PhD Workshop*. IFIP. June 2013, pp. 84–87.
- [147] Björn Richerzhagen, Dominik Stingl, Ronny Hans, Christian Groß, and Ralf Steinmetz. "Bypassing the Cloud: Peer-assisted Event Dissemination for Augmented Reality Games." In: *Proc. 14th IEEE Conference on Peer-to-Peer Computing (P2P)*. Sept. 2014, pp. 1–10.
- [148] Björn Richerzhagen, Dominik Stingl, Julius Rückert, and Ralf Steinmetz. "Simulator: Simulation and Prototyping Platform for Distributed Mobile Applications." In: *Proc. 8th International Conference on Simulation Tools and Techniques (SIMUTOOLS)*. ACM. Aug. 2015, pp. 99–108.

- [149] Björn Richerzhagen, Alexander Wagener, Nils Richerzhagen, Rhaban Hark, and Ralf Steinmetz. "A Framework for Publish/Subscribe Protocol Transitions in Mobile Crowds." In: *Proc. 10th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*. IFIP. June 2016, pp. 1–14.
- [150] Björn Richerzhagen, Stefan Wilk, Julius Rückert, Denny Stohr, and Wolfgang Effelsberg. "Transitions in Live Video Streaming Services." In: *Proc. Workshop on Design, Quality and Deployment of Adaptive Video Streaming (VideoNEXT)*. ACM. 2014, pp. 37–38.
- [151] Björn Richerzhagen, Julian Wulfheide, Heinz Koepl, Andreas Mauthe, Klara Nahrstedt, and Ralf Steinmetz. "Enabling Crowdsourced Live Event Coverage with Adaptive Collaborative Upload Strategies." In: *Proc. 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2016, pp. 1–3.
- [152] 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: *Proc. 40th IEEE Conference on Local Computer Networks (LCN)*. IEEE. 2015, pp. 462–465.
- [153] Nils Richerzhagen, Björn Richerzhagen, Rhaban Hark, Dominik Stingl, and Ralf Steinmetz. "Limiting the Footprint of Monitoring in Dynamic Scenarios through Multi-dimensional Offloading." In: *Proc. 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2016, pp. 1–9.
- [154] Nils Richerzhagen, Björn Richerzhagen, Michal Lipinski, Markus Weckesser, Roland Kluge, Rhaban Hark, and Ralf Steinmetz. "Exploring Transitions in Mobile Network Monitoring in Highly Dynamic Environments." In: *Demonstrations of the 41st IEEE Conference on Local Computer Networks (LCN-Demos)*. Nov. 2016, pp. 1–3.
- [155] Nils Richerzhagen, Björn Richerzhagen, Dominik Stingl, and Ralf Steinmetz. "The Human Factor: A Simulation Environment for Networked Mobile Social Applications." In: *Proc. International Conference on Networked Systems (NetSys)*. IEEE. Mar. 2017, pp. 1–8.
- [156] Nils Richerzhagen, Björn Richerzhagen, Michael Walter, Dominik Stingl, and Ralf Steinmetz. "Buddies, not Enemies: Fairness and Performance in Cellular Offloading." In: *Proc. 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. June 2016, pp. 1–9.
- [157] Nils Richerzhagen, Dominik Stingl, Björn Richerzhagen, Andreas Mauthe, and Ralf Steinmetz. "Adaptive Monitoring for Mobile Networks in Challenging Environments." In: *Proc. 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2015, pp. 1–8.
- [158] George F Riley and Thomas R Henderson. "The ns-3 network simulator." In: *Modeling and tools for network simulation (2010)*, pp. 15–34.

- [159] Elizabeth M Royer and Chai-Keong Toh. "A review of current routing protocols for ad hoc mobile wireless networks." In: *IEEE personal communications* 6.2 (1999), pp. 46–55.
- [160] Julius Rückert. "Large-scale Live Video Streaming over the Internet – Efficient and Flexible Content Delivery Using Network and Application-Layer Mechanisms." PhD thesis. TU Darmstadt, 2016.
- [161] Julius Rückert, Björn Richerzhagen, Eduardo Lidanski, Ralf Steinmetz, and David Hausheer. "TopT: Supporting Flash Crowd Events in Hybrid Overlay-based Live Streaming." In: *Proc. 14th IFIP Networking Conference (Networking)*. IEEE. 2015, pp. 1–9.
- [162] Patricia Ruiz and Pascal Bouvry. "Survey on Broadcast Algorithms for Mobile Ad Hoc Networks." In: *ACM Computing Surveys (CSUR)* 48.1 (2015), p. 8.
- [163] Enrique Saurez, Kirak Hong, Dave Lillethun, Umakishore Ramachandran, and Beate Ottenwälder. "Incremental deployment and migration of geo-distributed situation awareness applications in the fog." In: *Proc. ACM International Conference on Distributed and Event-based Systems (DEBS)*. ACM. 2016, pp. 258–269.
- [164] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Vito Latora. "Distance Matters: Geo-social Metrics for Online Social Networks." In: *WOSN*. 2010.
- [165] Jochen Schiller and Agnès Voisard. *Location-based Services*. Elsevier, 2004.
- [166] Bill Segall and David Arnold. "Elvin has left the building: A publish/subscribe notification service with quenching." In: *Proceedings of AUUG. Brisbane, Australia* (1997), pp. 243–255.
- [167] Jon Siegel and Dan Frantz. *CORBA 3 fundamentals and programming*. Vol. 2. John Wiley & Sons New York, NY, USA, 2000.
- [168] Thirunavukkarasu Sivaharan, Gordon Blair, and Geoff Coulson. "Green: A configurable and re-configurable publish-subscribe middleware for pervasive computing." In: *Proc. International Conferences On the Move to Meaningful Internet Systems (OTM)*. Springer. 2005, pp. 732–749.
- [169] Carl-Fredrik Sørensen, Maomao Wu, Thirunavukkarasu Sivaharan, Gordon S Blair, Paul Okanda, Adrian Friday, and Hector Duran-Limon. "A context-aware middleware for applications in mobile ad hoc environments." In: *Proc. Workshop on Middleware for Pervasive and Ad-hoc Computing (MPAC)*. ACM. 2004, pp. 107–110.
- [170] Eduardo Souto, Germano Guimarães, Glauco Vasconcelos, Mardoqueu Vieira, Nelson Rosa, Carlos Ferraz, and Judith Kelner. "Mires: a publish/subscribe middleware for sensor networks." In: *Personal and Ubiquitous Computing* 10.1 (2006), pp. 37–44.

- [171] Michael Stein, Alexander Frömmgen, Roland Kluge, Frank Löffler, Andy Schürr, Alejandro Buchmann, and Max Mühlhäuser. "TARL: Modeling topology adaptations for networking applications." In: *Proc. International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. ACM. 2016, pp. 57–63.
- [172] Michael Stein, Roland Kluge, Dario Mirizzi, Stefan Wilk, Andy Schürr, and Max Mühlhäuser. "Transitions on multiple layers for scalable, energy-efficient and robust wireless video streaming." In: *Proc. International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE. 2016, pp. 1–3.
- [173] Ralf Steinmetz, Melanie Holloway, Boris Koldehofe, Björn Richerzhagen, and Nils Richerzhagen. "Towards Future Internet Communications – Role of Scalable Adaptive Mechanisms." In: *ACADEMIA EUROPAEA* (2015), pp. 59–61.
- [174] Ralf Steinmetz and Klaus Wehrle. "What Is This Peer-to-Peer About?" In: *Peer-to-Peer Systems and Applications*. Springer, 2005, pp. 9–16.
- [175] Dominik Stingl. "Decentralized Monitoring in Mobile Ad Hoc Networks – Provisioning of Accurate and Location-Aware Monitoring Information." PhD thesis. TU Darmstadt, 2014.
- [176] Dominik Stingl, Christian Gross, Julius Rückert, Leonhard Nobach, Aleksandra Kovacevic, and Ralf Steinmetz. "Peerfactsim. kom: A simulation framework for peer-to-peer systems." In: *Proc. International Conference on High Performance Computing and Simulation (HPCS)*. IEEE. 2011, pp. 577–584.
- [177] Dominik Stingl, Björn Richerzhagen, Fabio Zöllner, Christian Gross, and Ralf Steinmetz. "PeerfactSim.KOM: Take it back to the Streets." In: *Proc. International Conference on High Performance Computing and Simulation (HPCS)*. IEEE. 2013, pp. 80–86.
- [178] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet applications." In: *ACM SIGCOMM Computer Communication Review* 31.4 (2001), pp. 149–160.
- [179] Andrew Tanenbaum. *Computer Networks*. 4th. Prentice Hall Professional Technical Reference, 2002. ISBN: 0130661023.
- [180] Sasu Tarkoma. *Publish/subscribe systems: design and principles*. John Wiley & Sons, 2012.
- [181] Wesley W Terpstra, Stefan Behnel, Ludger Fiege, Andreas Zeidler, and Alejandro P Buchmann. "A peer-to-peer approach to content-based publish/subscribe." In: *Proc. International Workshop on Distributed Event-based Systems (DEBS)*. ACM. 2003, pp. 1–8.
- [182] Wesley W Terpstra, Jussi Kangasharju, Christof Leng, and Alejandro P Buchmann. "Bubblestorm: resilient, probabilistic, and exhaustive peer-to-peer search." In: *ACM SIGCOMM Computer Communication Review*. Vol. 37. 4. ACM. 2007, pp. 49–60.

- [183] Peter Triantafillou and Andreas Economides. "Subscription summarization: A new paradigm for efficient publish/subscribe systems." In: *Proc. International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2004, pp. 562–571.
- [184] Glen Van Brummelen. *Heavenly mathematics: The forgotten art of spherical trigonometry*. Princeton University Press, 2013.
- [185] Wouter G Van Toll, Atlas F Cook, and Roland Geraerts. "Real-time density-based crowd simulation." In: *Computer Animation and Virtual Worlds* 23.1 (2012), pp. 59–69.
- [186] Long Vu, Phuong Nguyen, Klara Nahrstedt, and Björn Richerzhagen. "Characterizing and Modeling People Movement from Mobile Phone Sensing Traces." In: *Pervasive and Mobile Computing* 17 (2015), pp. 220–235.
- [187] Cong Wang, Amr Rizk, and Michael Zink. "Squad: A spectrum-based quality adaptation for dynamic adaptive streaming over HTTP." In: *Proc. International Conference on Multimedia Systems (MMSys)*. ACM. 2016, p. 1.
- [188] Haiquan Wang, Tao Zhu, Shuo Lei, Wenjing Yang, and Yu Wang. "CMTG: A content-based mobile tendency geocast routing protocol in urban vehicular networks." In: *International Journal of Distributed Sensor Networks* (2015).
- [189] Markus Weckesser, Malte Lochau, Thomas Schnabel, Björn Richerzhagen, and Andy Schürr. "Mind the Gap! Automated Anomaly Detection for Potentially Unbounded Cardinality-based Feature Models." In: *Fundamental Approaches to Software Engineering*. Springer Berlin Heidelberg, 2016, pp. 158–175.
- [190] Marc Werner, Johannes Schwandke, Matthias Hollick, Oliver Hohlfeld, Torsten Zimmermann, and Klaus Wehrle. "STEAN: A Storage and Transformation Engine for Advanced Networking Context." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IFIP. 2016, pp. 91–99.
- [191] Matthias Wichtlhuber, Björn Richerzhagen, Julius Rückert, and David Hausheer. "TRANSIT: Supporting Transitions in Peer-to-Peer Live Video Streaming." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IEEE. 2014, pp. 1–9.
- [192] Wikipedia. *Transition (computer science)* — *Wikipedia, The Free Encyclopedia*. 2016. URL: [https://en.wikipedia.org/wiki/Transition_\(computer_science\)](https://en.wikipedia.org/wiki/Transition_(computer_science)).
- [193] Stefan Wilk. "Quality-aware Content Adaptation in Digital Video Streaming." PhD thesis. TU Darmstadt, 2016.
- [194] Stefan Wilk, Julius Rückert, Denny Stohr, Björn Richerzhagen, and Wolfgang Effelsberg. "Efficient Video Streaming through Seamless Transitions Between Unicast and Broadcast." In: *Proc. 2nd Conference on Networked Systems (NetSys)*. 2015, pp. 1–2.
- [195] Stefan Wilk, Roger Zimmermann, and Wolfgang Effelsberg. "Leveraging transitions for the upload of user-generated mobile video." In: *Proc. International Workshop on Mobile Video*. ACM. 2016, p. 5.

- [196] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. "Identifying Diverse Usage Behaviors of Smartphone Apps." In: *Proc. ACM SIGCOMM Conference on Internet Measurement (IM)*. ACM. 2011, pp. 329–344.
- [197] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. "LCARS: a location-content-aware recommender system." In: *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2013, pp. 221–229.
- [198] Eiko Yoneki and Jean Bacon. "An Adaptive Approach to Content-Based Subscription in Mobile Ad Hoc Networks." In: *Percom Workshops*. Citeseer. 2004, pp. 92–97.
- [199] Sanghyun Yoo, Jin Hyun Son, and Myoung Ho Kim. "A scalable publish/-subscribe system for large mobile ad hoc networks." In: *Journal of Systems and Software* 82.7 (2009), pp. 1152–1162.
- [200] Minghe Yu, Guoliang Li, Ting Wang, Jianhua Feng, and Zhiguo Gong. "Efficient filtering algorithms for location-aware publish/subscribe." In: *IEEE Transactions on Knowledge and Data Engineering* 27.4 (2015), pp. 950–963.
- [201] Farid Zaid, Parag S Mogre, Andreas Reinhardt, Diego Costantini, and Ralf Steinmetz. "iVu. KOM: A Framework for Viewer-centric Mobile Location-based Services." In: *PIK-Praxis der Informationsverarbeitung und Kommunikation* 33.4 (2010), pp. 284–290.
- [202] Andreas Zeidler. "A distributed publish/subscribe notification service for pervasive environments." PhD thesis. TU Darmstadt, 2005.
- [203] Dongxiang Zhang, Long Guo, Liqiang Nie, Jie Shao, Sai Wu, and Heng Tao Shen. "Targeted Advertising in Public Transportation Systems with Quantitative Evaluation." In: *ACM Transactions on Information Systems (TOIS)* 35.3 (2017), p. 20.
- [204] Yinghua Zhou, Xing Xie, Chuang Wang, Yuchang Gong, and Wei-Ying Ma. "Hybrid index structures for location-based web search." In: *Proc. ACM International Conference on Information and Knowledge Management*. ACM. 2005, pp. 155–162.
- [205] Michael Zink. *Scalable video on demand: adaptive internet-based distribution*. John Wiley & Sons, 2013.
- [206] Michael Zink, Jens Schmitt, and Ralf Steinmetz. "Layer-encoded video in scalable adaptive streaming." In: *IEEE Transactions on Multimedia* 7.1 (2005), pp. 75–84.
- [207] Michael Zink, Kyoungwon Suh, Yu Gu, and Jim Kurose. "Watch global, cache local: YouTube network traffic at a campus network: measurements and implications." In: *Electronic Imaging 2008*. International Society for Optics and Photonics. 2008, pp. 681805–681805.

- [208] Michael Zink, Kyoungwon Suh, Yu Gu, and Jim Kurose. “Characteristics of YouTube network traffic at a campus network—measurements, models, and implications.” In: *Computer networks* 53.4 (2009), pp. 501–514.

All web pages cited in this work have been checked in May 2017. However, due to the dynamic nature of the World Wide Web, their long-term availability cannot be guaranteed.

APPENDIX

A.1 EXTENDED STUDY OF FILTER SCHEMES

This section provides additional evaluation results for the performance and cost characteristics of the individual filter schemes integrated into `BYPASS.KOM` (as discussed in Section 4.2.4). The evaluation is conducted in the same setup as described in Section 6.1 and results are reported for the `DA` mobility model and the `AR` workload, unless otherwise noted. In the following, we compare the performance and cost of individual filter schemes, discuss the influence of scheme-specific configuration parameters and, finally, evaluate the impact of reduced location accuracy.

Performance and Cost of Individual Filter Schemes

Figure 65 shows the recall, precision, and traffic caused by the utilization of the respective filter scheme. We compare the characteristics under different workloads, with the workload models being discussed in detail in Section 6.1.2. The download traffic resulting from a scheme's utilization is directly tied to the precision of the scheme. A higher precision results in a reduced number of unnecessarily transmitted events. Consequently, the channel-based schemes `GRID` and `eGRID` cause significantly higher traffic than `RADIAL` and `STE` under all workload models. Compared to `RADIAL` as the scheme with the highest precision (0.98 on average), `GRID` with an average precision between 0.1 and 0.2 causes about four times higher traffic.

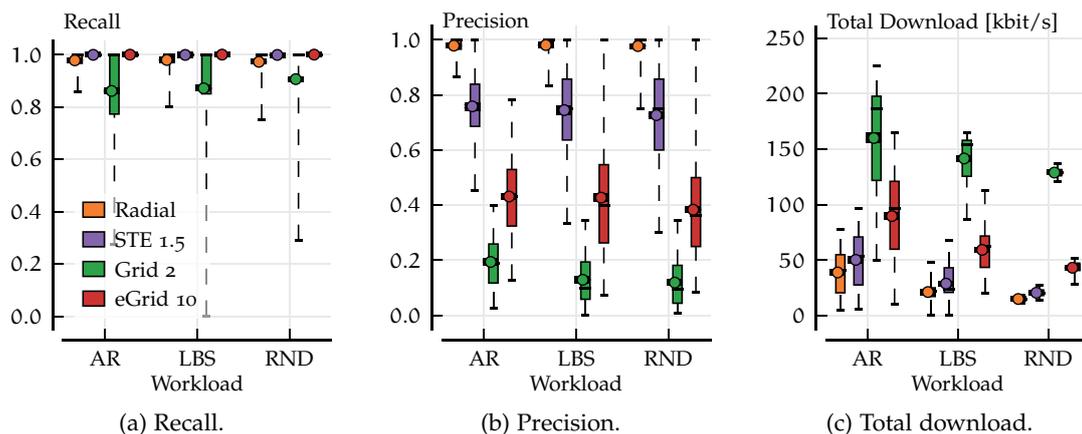


Figure 65: Performance of the individual filter schemes.

`STE` and `eGRID` both maintain a constant recall of 1.0. `STE` outperforms `RADIAL` in terms of recall, as it deals with mobility by intentionally extending the subscribed

area in the direction of movement, as described in Section 4.2.4. This has an impact on the precision that is decreased to about 0.75. Still, the traffic caused by STE is only slightly higher than the traffic for RADIAL. This is due to the fact that the size of the extension shape defined in STE depends on a client's movement speed. If a client is not moving, STE behaves just like RADIAL, leading to high recall and precision and, consequently, low traffic.

Comparing the behavior for different workloads in Figure 65, we can observe that the results for the AR workload model are more skewed than the results reported for LBS and RND. This is caused by the heterogeneous load resulting from the AR model. In AR, load directly depends on the current location of clients and the resulting client density. The density is higher nearby attraction points compared to other areas, leading to the skewed results.

Impact of the Space-Time Envelope Size

The STE filter scheme can be configured with a parameter α that determines the size of the extension shape relative to the client's current movement speed. As introduced in Section 4.2.4, the length of the extension shape is calculated as $\alpha \cdot 60 \cdot |\vec{s}_u|$, with $|\vec{s}_u|$ being the client's movement speed in meters per second. In this section, we assess the performance and cost characteristics that result from different values of α . We vary α from 0.5 to 2.5 in 0.5 increments. Recall, precision, and the resulting traffic for different values of α are shown in Figure 66.

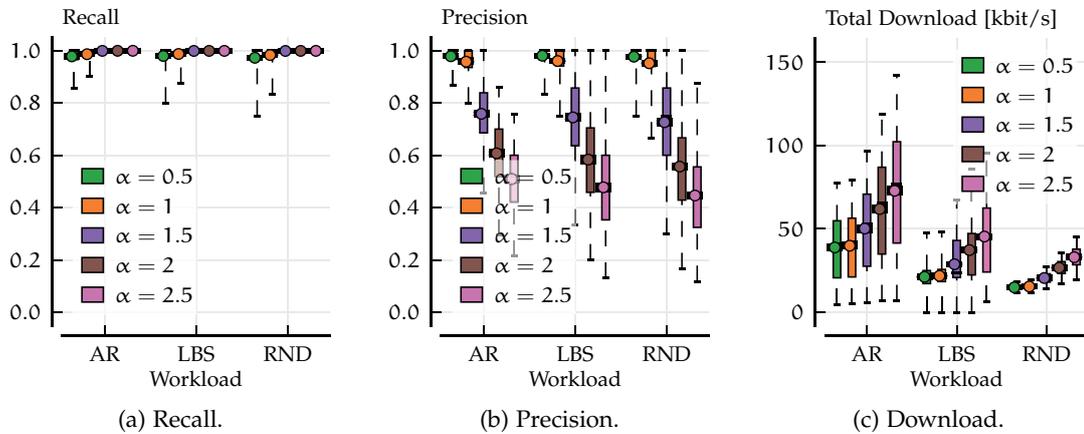


Figure 66: Impact of α on the performance of STE.

As expected, the recall of STE for $\alpha < 1$ approaches that of the RADIAL filter scheme. In the extreme case of $\alpha = 0$, one would observe the same filtering behavior as for RADIAL. The recall quickly saturates at 1.0 for STE with $\alpha \geq 1.5$. Higher values of α only lead to decreases precision and, consequently, increased traffic. We chose a default value of $\alpha = 1.5$ for our evaluations, given that this configuration leads to perfect recall of 1.0 at a reasonable decrease of precision to about 0.75.

Impact of the Grid Factor on Channel-based Filter Schemes

The performance of the grid-based filter schemes GRID and eGRID further depends on how well the chosen grid structure can reflect the subscriptions issued by clients' applications. In the following, we vary the *grid factor* introduced in Section 4.2.4. The larger the respective factor, the more cells are created. A grid factor of 2 leads to 4 channels, each covering a square area with an edge length of 650 m in our evaluation setup. With a grid factor of 20, we end up with 400 channels, each channel responsible for a square area with 65 m edge length. The radius of the AoI an application subscribes to is fixed to 125 m, as discussed in Section 6.1.

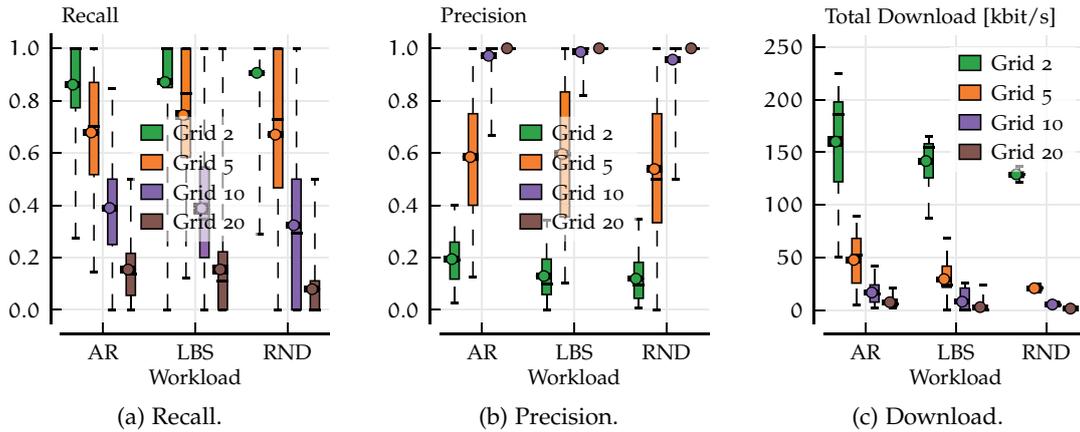


Figure 67: Impact of different grid sizes on the performance of GRID.

The results for GRID are reported in Figure 67. As clients only subscribe to one channel in GRID, the recall decreases quickly for larger grid factors. In order to achieve a reasonably high recall of above 0.85 for all workload models, GRID needs to be configured with a grid factor of 2. However, this leads to a bad precision of below 0.2 on average. The traffic caused by GRID for a reasonably high recall exceeds that of all other filter schemes as already shown in Figure 65.

With eGRID, clients subscribe to all channels that fit within the MBR of their subscriptions, as discussed in Section 4.2.4. Consequently, the scheme is able to maintain a perfect recall of 1.0 under all workload models. The precision is directly determined by the size of the grid cells. For smaller cells (i. e., for a larger grid factor), the resulting MBR matches the actual AoI of a client more closely, leading to higher precision and, consequently, reduced traffic. At the same time, the coordination overhead due to more frequent channel reassignments increases (not shown here).

Given that eGRID relies on the MBR to assign channels to clients, the theoretical upper bound for its precision with respect to a circular AoI is $2/\pi \approx 0.64$. In the ideal case, the MBR corresponds to a square with edge length $2r$ (with r being the radius of the circular AoI) that is centered around the location of the client. In this case (and under the assumption of a sufficiently large number of events being published to equally distributed locations) the precision equals the ratio of both areas:

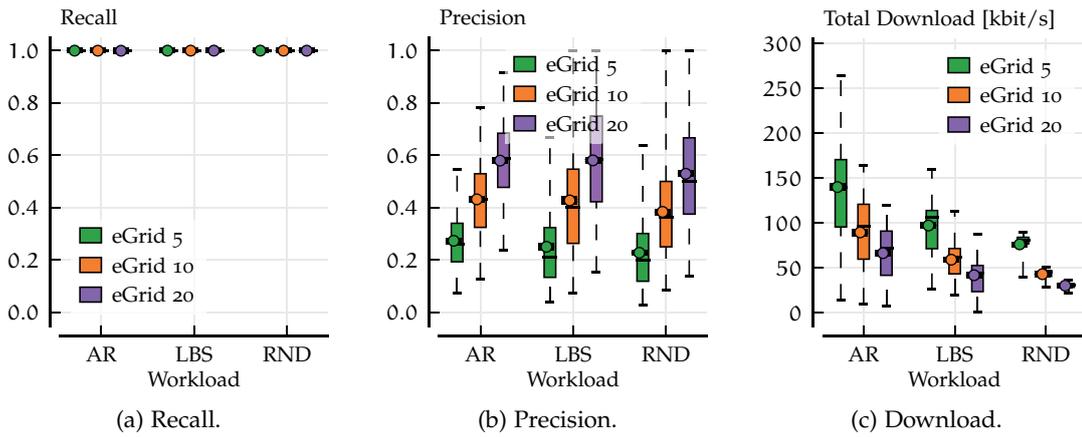


Figure 68: Impact of different grid sizes on the performance of eGRID.

$(4r^2)/(2\pi r^2) = 2/\pi \approx 0.64$. We observe higher values for the precision of a fraction of our publications in Figure 68, which is caused by the non-equal distribution of target locations in our workload models and the (comparably) small number of events.

Impact of Location Accuracy on Filter Schemes

In our evaluation, a new location is reported by the mobile OS every 5 seconds. Depending on the realization of the respective filter scheme, this updated location is sent to the broker (e.g., in RADIAL and STE) or used to check the current channel assignment locally (e.g., in GRID and eGRID). Within this evaluation, we vary the location update interval to evaluate the robustness of the respective filter schemes to outdated location information.

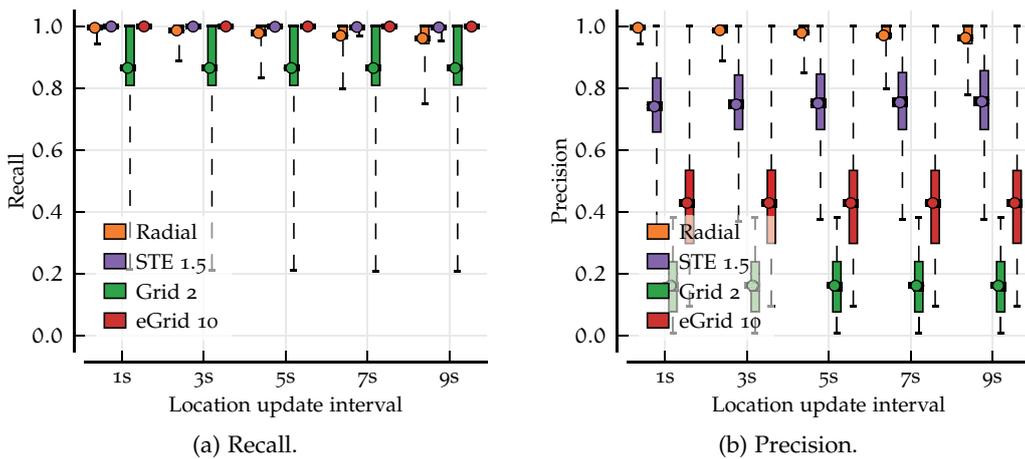


Figure 69: Recall and precision for different location update intervals.

Figure 69 reports the recall and precision for different filter schemes depending on the location update interval. The channel-based filter schemes GRID and eGRID are robust to outdated location information given that the respective channels already cover an area exceeding the AoI of a client. For the interactive mobile social applications considered in our work, it is a reasonable assumption that updated location information is available every few seconds due to the utilization of GPS. In less interactive applications, longer location update intervals can lead to a performance degradation even for channel-based schemes.

RADIAL clearly suffers from reduced location accuracy, given that the average recall drops from around 1 for a location update interval of 1 second to about 0.95 for an interval of 9 seconds. Outdated location information further leads to a decrease in the precision of RADIAL. STE maintains an average recall of 1.0, given that the extension shape is designed to deal with continued client mobility if the movement direction remains similar. For a fraction of clients this assumption does not hold, leading to slightly decreased recall reported for the 2.5th percentile.

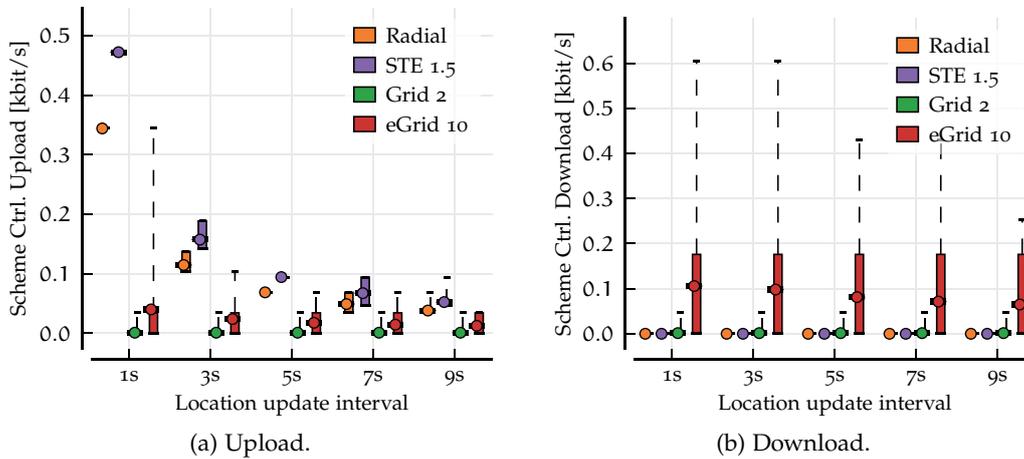


Figure 70: Client-side traffic caused by different location update intervals.

Figure 70 reports the traffic caused by the filter schemes' context update protocols. While the absolute figures are well below 0.5 kbit/s, their relative comparison allows us to assess how the respective filter schemes deal with location updates available on mobile clients. Given the large area covered by a channel in GRID with a grid factor of 2, the filter scheme's context update protocol is not affected by the considered location update intervals, as discussed previously. For channels covering a smaller area as considered in eGRID, less frequent location updates lead to a slight decrease in the number of reassignment requests sent to the broker. Consequently, the upload traffic caused by the context update protocol decreases slightly. Due to less assignment requests being issued by clients, the broker sends slightly less reassignment messages to clients, leading to a reduction in the download traffic caused by the context update protocol.

Clients using RADIAL and STE report their current location whenever updated information is available. Consequently, the upload traffic decreases with longer location

update intervals. In both schemes, the broker does not send any messages via the context update protocol, leading to a download traffic of 0 kbit/s. Messages issued by clients using STE are slightly larger than those issued by the RADIAL scheme. This is due to the fact that the current movement vector is included in the message to enable calculation of the extension shape at the broker, as discussed in Section 4.2.4.

A.2 PARTIAL TRANSITIONS BETWEEN FILTER SCHEMES

We compare the self-adaptive system MULTI against the execution of partial transitions between the ATTRACT and RADIAL filter schemes based on the location of attraction points. Augmenting the discussion in Section 6.2.4, we present the evaluation results for the MA mobility model in the following. We include the previously reported results for the DA mobility model for comparison. The overall characteristics in terms of recall, precision, and cellular traffic are shown in Figure 71.

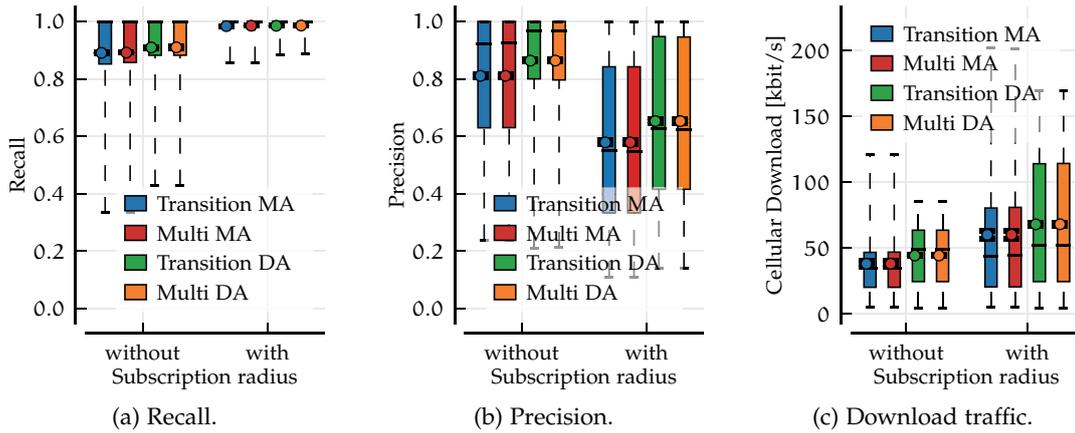


Figure 71: Performance and cost comparison.

For both mobility models (and the corresponding attraction points), including the subscription radius to extend the area covered by a channel in ATTRACT leads to a recall of approximately 0.99. The MA mobility model defines more attraction points than DA. This increases the area around attraction points where clients suffer from decreased recall if the radius of subscriptions is not taken into account, as shown in Figure 72a. Consequently, the average recall in the MA scenario reaches 0.88 compared to approximately 0.9 for DA, if radii are not considered. Additionally, the fraction of the simulated area that is covered by channels is higher than for the DA scenario, resulting in an overall decrease in precision.

As previously reported for DA, including the radius of subscriptions when matching against the area defined by a channel in ATTRACT leads to a recall of approximately 1.0. This comes at the cost of reduced accuracy and, consequently, increased cellular download traffic, as reported in Figure 71c. The spatial distribution of recall and precision for the MA mobility model is shown in Figure 72 for completeness.

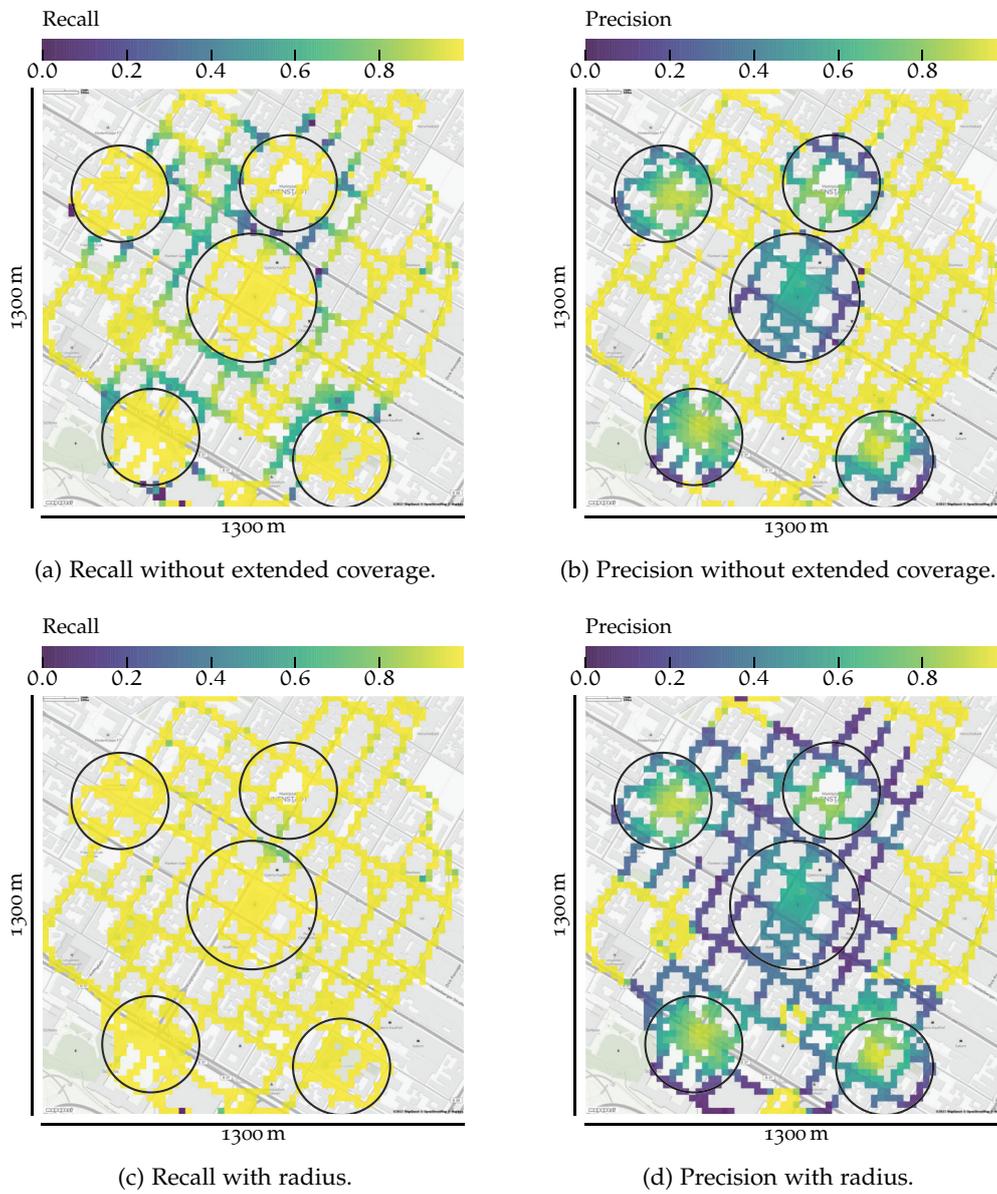


Figure 72: Performance with and without extended channel coverage.

A.3 EXTENDED STUDY OF DISSEMINATION MECHANISMS

In this section, we present additional evaluation results that augment our evaluation of dissemination mechanisms presented in Section 6.3. We study the behavior of individual local dissemination mechanisms under different client densities. Further, we discuss the impact of the forwarding probability p on the performance of the Gossip mechanism. The setup of the respective evaluations remains as described in Section 6.1, unless otherwise noted.

Scalability of Local Dissemination Mechanisms

To assess the behavior of ad hoc dissemination mechanisms under different client densities, we utilize a ramp-based churn model as proposed in [13]. The number of active clients is increased linearly until it reaches a total of 300 clients after 25 minutes. We include the number of active clients in the plots presented in the following. All results are reported for the AR workload model. Consequently, the dissemination mechanisms are subject to geofencing, limiting forwarding of a local event to actual subscribers of the respective event, as discussed in Section 4.3.1.

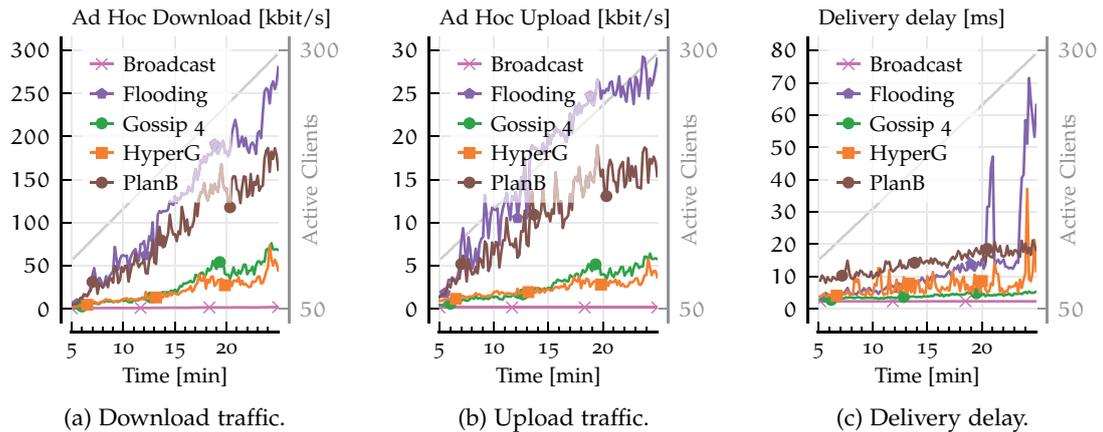


Figure 73: Behavior of dissemination schemes with increasing client density, AR workload.

Figure 73 shows the traffic characteristics of individual dissemination mechanisms with increasing number of clients. Additionally, the resulting mean dissemination delay for the distribution of events is shown. Regarding traffic, the aggressive mechanisms FLOODING and PLAN-B cause a significant increase for larger numbers of clients and, consequently, denser networks. GOSSIP and HYPERG exhibit roughly the same characteristics, causing only a third of the traffic of PLAN-B and approximately a fifth of the traffic caused by FLOODING. Traffic caused by the one-hop dissemination mechanism BROADCAST is negligible in comparison. The delivery delay remains below 20 ms for all mechanisms, with the exception of FLOODING. FLOODING overloads the network, leading to increasing queuing times and, consequently, an increased delivery delay. For HYPERG, a slight increase in delivery delay can be observed to-

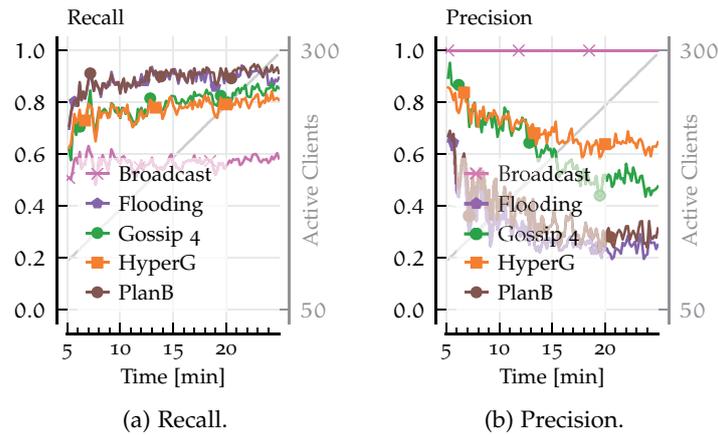


Figure 74: Behavior of dissemination schemes with increasing client density, AR workload.

wards the end of the simulated time. This is most likely caused by the retransmission mechanism that is used in HYPERG to deal with intermittent network partitions.

The recall of the dissemination schemes saturates quickly at approximately 0.9 for PLAN-B and FLOODING, as reported in Figure 74. The less aggressive schemes GOSSIP and HYPERG reach a recall of between 0.8 and 0.9, depending on the client density. With the AR workload about 60% of subscribers are reached by single-hop broadcasting. The precision of all multi-hop dissemination schemes decreases with increasing client density. Single-hop dissemination with BROADCAST is not affected, as the transmission radius resulting from the network model configuration discussed in Section 6.1.3 is smaller than the radius of a client's AoI.

Impact of the Forwarding Probability on the Performance of Gossiping

We rely on GOSSIP with a default forwarding probability $p = 0.4$ for our evaluation of coexisting mechanisms and hybrid dissemination mechanisms presented in Chapter 6. In this section, we briefly discuss the impact of the configuration parameter p on the performance and cost of the GOSSIP dissemination mechanism. We consider the impact of client density as well as different workload models.

Figure 75 reports the traffic resulting from different values of p . For $p = 1.0$, the mechanism behaves like FLOODING, as all messages are forwarded once by each client. As expected, lower values of p reduce the traffic significantly.

The resulting recall depends on the mobility model and the workload characteristics, as reported in Figure 75. For the AR workload, where events only need to be transmitted to nearby clients, a recall of 0.8 is achieved for $p = 0.4$. For larger values of p , the recall increases slightly, reaching 0.9 as previously reported for FLOODING. However, increasing the recall from 0.8 to 0.87 by changing p from 0.4 to 0.7 leads to a three times increase in traffic.

For other workloads, the impact of increasing p is more significant given that events need to traverse a potentially large amount of clients to reach their destina-

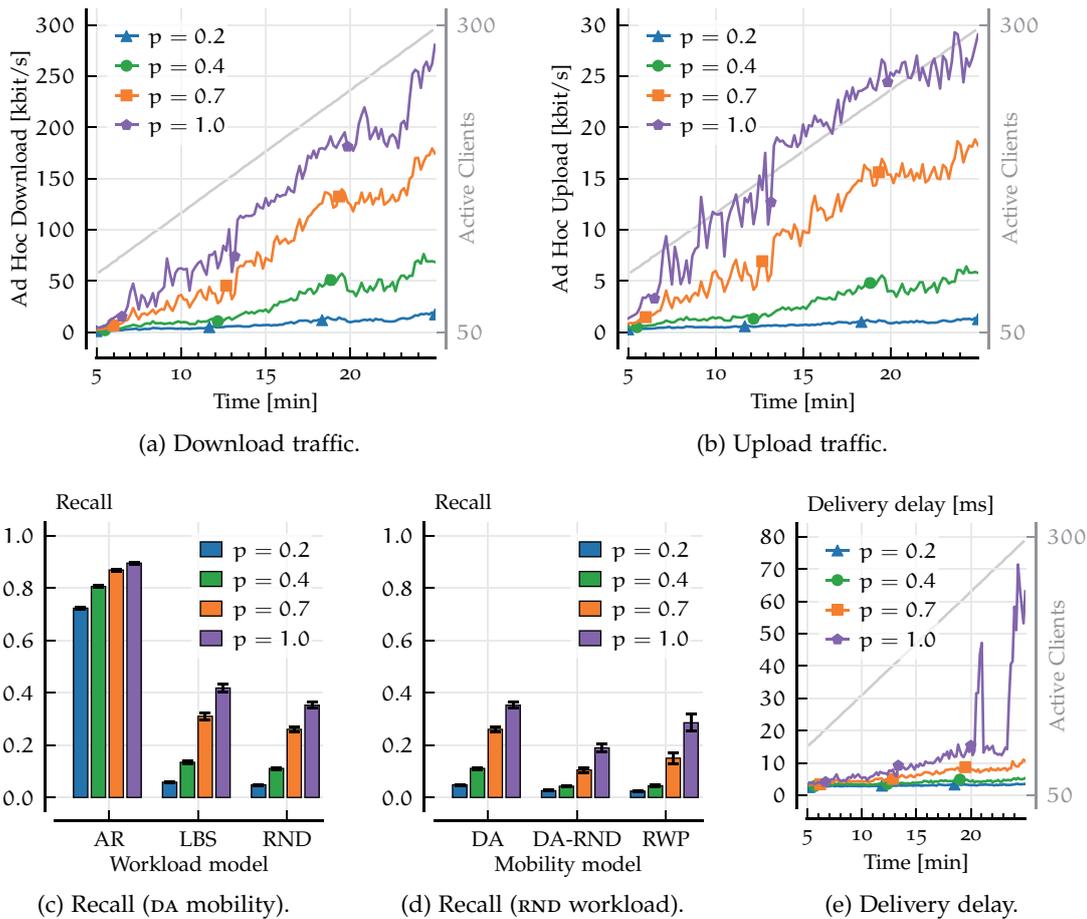


Figure 75: Impact of the forwarding probability p on the characteristics of Gossip.

tion. Even with $p = 1.0$, we only achieve a recall of 0.4 for the RND and LBS workload models. This behavior is largely independent from the utilized mobility model, as reported in Figure 75d for the RND workload. With more messages being forwarded by clients, the overall delivery delay of events increases slightly, as shown in Figure 75e. Gossip remains stable for $p \leq 0.7$, while overload situations occur for $p = 1.0$ as reported previously.

Impact of the Mobility Model on Hybrid Dissemination

We discussed hybrid event dissemination modes in Section 6.3.2, showing the achieved performance and the resulting offloading benefits. However, we limited the discussion to the DA mobility model for brevity. In this section, we discuss the impact of different mobility models on the hybrid dissemination of events, further extending the discussion in Section 6.3.2.

Figure 76 shows the performance of our hybrid modes compared to CLOUD and AD HOC. The hybrid dissemination modes AUGMENT and GATEWAY rely on the cloud

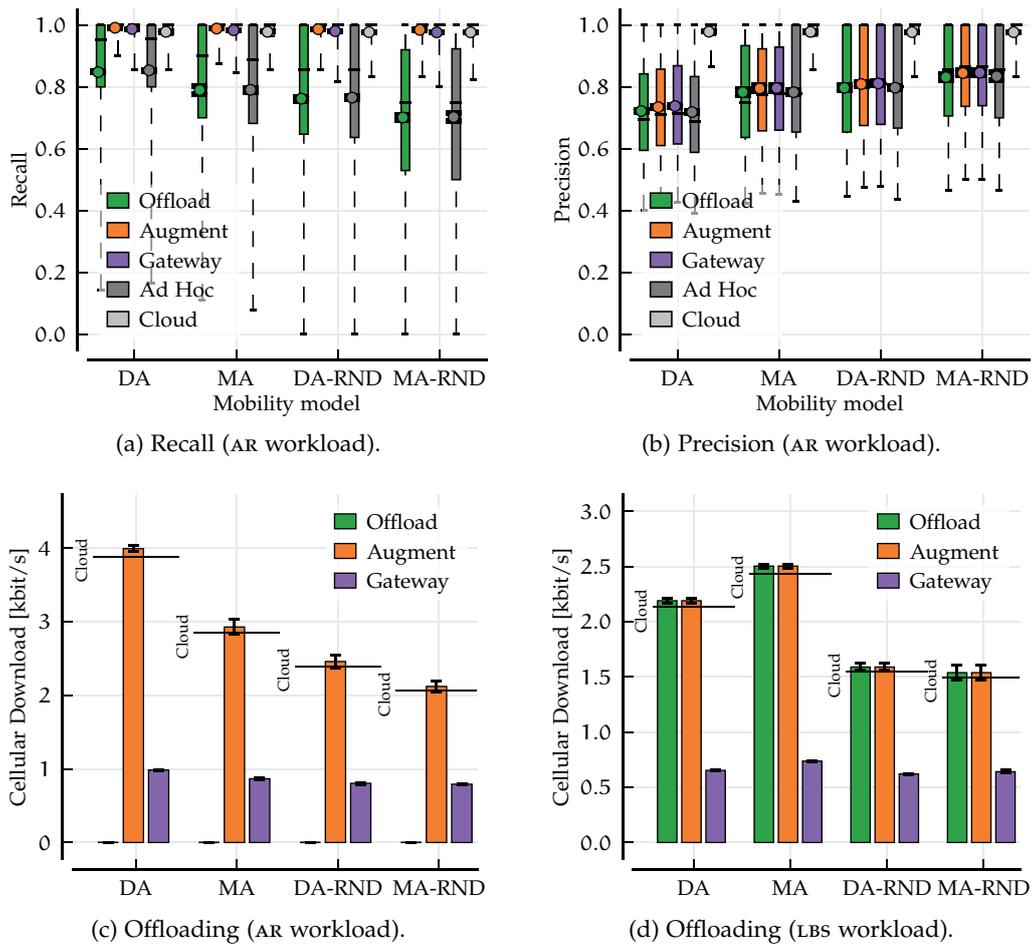


Figure 76: Hybrid dissemination under different mobility models (AR workload).

to ensure delivery of events, resulting in a constantly high recall of approximately 0.99. OFFLOAD suffers from the low recall of pure ad hoc event dissemination, leading to the respective recall of between 0.7 and 0.85 on average. The precision follows the same trend regardless of the mobility model.

However, the mobility model has an impact on the offloading effect achievable with the utilization of gateways. The offloading effect of the hybrid mode GATEWAY decreases if client mobility is no longer influenced by strong attraction points. Consequently, comparing DA and DA-RND, the offloading effect is reduced from approximately 75 % to 60 %. The same trend can be observed for the MA and MA-RND mobility models. For the LBS workload, the effect is less severe but follows the same trend.

A.4 LIST OF ACRONYMS

AoI	Area Of Interest
API	Application Programming Interface
AS	Autonomous System
CEP	Complex Event Processing
CTS	Clear To Send
DHT	Distributed Hash Table
DTN	Delay-tolerant Networking
GPS	Global Positioning System
IGMP	Internet Group Management Protocol
IP	Internet Protocol
ISO	Organization For Standardization
LTE	Long Term Evolution
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
MBR	Minimum Bounding Rectangle
MLD	Multicast Listener Discovery
NAT	Network Address Translation
NVE	Networked Virtual Environment
OS	Operating System
OSI	Open Systems Interconnection
OSM	OpenStreetMap
P2P	Peer-to-Peer
PDU	Protocol Data Unit
PoI	Point Of Interest
QoE	Quality Of Experience
QoS	Quality Of Service
RTS	Request To Send
SCTP	Stream Control Transmission Protocol
SDN	Software-defined Network
SDWN	Software-defined Wireless Network
SIS	State Information System
STE	Space-Time Envelope
TCP	Transmission Control Protocol
TTL	Time To Live

UDP	User Datagram Protocol
UUID	Universally Unique Identifier
VANET	Vehicular Ad Hoc Network
VoD	Video On Demand
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

A.5 SUPERVISED STUDENT THESES

- [1] Björn Büschke. "Adaptive Gateway Selection Strategies for Event Dissemination in Mobile Applications." Bachelor Thesis. TU Darmstadt, 2016.
- [2] Dennis Drexler. "Augmenting Internet-wide Publish/Subscribe Systems with Software Defined Networking Principles." Master Thesis. TU Darmstadt, 2015.
- [3] Lukas Fey. "Context-aware Ad Hoc Event Dissemination Protocols for Mobile Augmented Reality Games." Bachelor Thesis. TU Darmstadt, 2015.
- [4] Leo Fuhr. "Cellular-assisted Publish/Subscribe for Mobile Ad Hoc Networks." Master Thesis. TU Darmstadt, 2013.
- [5] Thorben Hehmann. "Dynamic Provisioning of Cloudlet-based Brokers for Location-aware Publish/Subscribe." Master Thesis. TU Darmstadt, 2015.
- [6] Marc Schiller. "Prototypical Realization of Transition-enabled Ad Hoc Communication for a Mobile Pervasive Game." Bachelor Thesis. TU Darmstadt, 2015.
- [7] Sophie Schönherr. "Context-based Adaptation of Publish/Subscribe Semantics for Mobile Applications." Bachelor Thesis. TU Darmstadt, 2015.
- [8] Christoph Storm. "Collaborative Location Retrieval under Data Uncertainty." Master Thesis, co-supervised. TU Darmstadt, 2017.
- [9] Alexander Wagener. "Supporting Transitions between Publish/Subscribe Mechanisms in Mobile Ad Hoc Networks." Master Thesis. TU Darmstadt, 2013.
- [10] Julian Wulfheide. "Design of Collaborative Upload Strategies for Crowdsourced Live Event Coverage." Master Thesis. TU Darmstadt, 2016.
- [11] Julian Zobel. "Transitions between Subscription Schemes for Location-based Mobile Services." Bachelor Thesis. TU Darmstadt, 2016.

AUTHOR'S PUBLICATIONS

MAIN PUBLICATIONS

- [1] Björn Richerzhagen, Nils Richerzhagen, Sophie Schönherr, Rhaban Hark, and Ralf Steinmetz. "Stateless Gateways – Reducing Cellular Traffic for Event Distribution in Mobile Social Applications." In: *Proc. 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. Aug. 2016, pp. 1–9.
- [2] Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe, and Ralf Steinmetz. "Seamless Transitions Between Filter Schemes for Location-based Mobile Applications." In: *Proc. 41st IEEE Conference on Local Computer Networks (LCN)*. Nov. 2016, pp. 1–9.
- [3] Björn Richerzhagen and Ralf Steinmetz. "Towards an Adaptive Publish/Subscribe Approach Supporting Transitions." In: *Proc. 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS), PhD Workshop*. IFIP. June 2013, pp. 84–87.
- [4] Björn Richerzhagen, Dominik Stingl, Ronny Hans, Christian Groß, and Ralf Steinmetz. "Bypassing the Cloud: Peer-assisted Event Dissemination for Augmented Reality Games." In: *Proc. 14th IEEE Conference on Peer-to-Peer Computing (P2P)*. Sept. 2014, pp. 1–10.
- [5] Björn Richerzhagen, Dominik Stingl, Julius Rückert, and Ralf Steinmetz. "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications." In: *Proc. 8th International Conference on Simulation Tools and Techniques (SIMUTOOLS)*. ACM. Aug. 2015, pp. 99–108.
- [6] Björn Richerzhagen, Alexander Wagener, Nils Richerzhagen, Rhaban Hark, and Ralf Steinmetz. "A Framework for Publish/Subscribe Protocol Transitions in Mobile Crowds." In: *Proc. 10th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*. IFIP. June 2016, pp. 1–14.
- [7] Björn Richerzhagen, Julian Wulfheide, Heinz Koepl, Andreas Mauthe, Klara Nahrstedt, and Ralf Steinmetz. "Enabling Crowdsourced Live Event Coverage with Adaptive Collaborative Upload Strategies." In: *Proc. 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2016, pp. 1–3.
- [8] Nils Richerzhagen, Björn Richerzhagen, Dominik Stingl, and Ralf Steinmetz. "The Human Factor: A Simulation Environment for Networked Mobile Social Applications." In: *Proc. International Conference on Networked Systems (NetSys)*. IEEE. Mar. 2017, pp. 1–8.

- [9] Matthias Wichtlhuber, Björn Richerzhagen, Julius Rückert, and David Hausheer. "TRANSIT: Supporting Transitions in Peer-to-Peer Live Video Streaming." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IEEE. 2014, pp. 1–9.

CO-AUTHORED PUBLICATIONS

- [10] Paul Baumann, Johannes Klaus, Björn Richerzhagen, Wilhelm Kleiminger, and Silvia Santini. "The UbiDCC Data Set: Collecting Wi-Fi and Bluetooth Scans During a Large-scale Conference." In: *Proc. 1st Workshop on Context Sensing and Activity Recognition (CSAR)*. ACM. 2015, pp. 25–30.
- [11] Alexander Frömmgen, Björn Richerzhagen, Julius Rückert, David Hausheer, Ralf Steinmetz, and Alejandro Buchmann. "Towards the Description and Execution of Transitions in Networked Systems." In: *Proc. 9th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*. IFIP. June 2015, pp. 17–29.
- [12] Christian Groß, Fabian Kaup, Dominik Stingl, Björn Richerzhagen, David Hausheer, and Ralf Steinmetz. "EnerSim: an Energy Consumption Model for Large-Scale Overlay Simulators." In: *Proc. 28th IEEE Conference on Local Computer Networks (LCN)*. 2013, pp. 252–255.
- [13] Christian Groß, Björn Richerzhagen, and Max Lehn. "Structured Search Overlays." In: *Benchmarking Peer-to-Peer Systems*. Springer Berlin Heidelberg, 2013, pp. 49–67.
- [14] Christian Groß, Björn Richerzhagen, Dominik Stingl, Jens Weber, David Hausheer, and Ralf Steinmetz. "Geoswarm: A Multi-Source Download Scheme for Peer-to-Peer Location-based Services." In: *Proc. 13th IEEE International Conference on Peer-to-Peer Computing*. IEEE. 2013, pp. 1–10.
- [15] Christian Groß, Dominik Stingl, Christian Gottron, Björn Richerzhagen, Christoph Munker, and David Hausheer. *Harnessing Mobile Ad Hoc Communication for Decentralized Location-Based Services*. Tech. rep. Peer-to-Peer Systems Engineering Lab, TU Darmstadt, Germany, 2012.
- [16] Christian Groß, Dominik Stingl, Björn Richerzhagen, Andreas Hemel, Ralf Steinmetz, and David Hausheer. "Geodemlia: A Robust Peer-to-Peer Overlay Supporting Location-based Search." In: *Proc. 12th IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE. 2012, pp. 25–36.
- [17] Ronny Hans, David Steffen, Ulrich Lampe, Björn Richerzhagen, and Ralf Steinmetz. "Setting Priorities: A Heuristic Approach for Cloud Data Center Selection." In: *Proc. 5th International Conference on Cloud Computing and Services Science (CLOSER) (2015)*, pp. 221–228.
- [18] Rhaban Hark, Nils Richerzhagen, Björn Richerzhagen, Amr Rizk, and Ralf Steinmetz. "Towards an Adaptive Selection of Loss Estimation Techniques in Software-defined Networks." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IEEE. 2017, pp. 1–9.

- [19] Rhaban Hark, Amr Rizk, Nils Richerzhagen, Björn Richerzhagen, and Ralf Steinmetz. "Poster: Isolated In-Band Communication for Distributed SDN Controllers." In: *Proc. IFIP Networking Conference (IFIP Networking)*. IEEE. 2017, pp. 1–2.
- [20] Björn Richerzhagen, Boris Koldehofe, and Ralf Steinmetz. "Immense Dynamism." In: *German Research* 37.2 (2015), pp. 24–27.
- [21] 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: *Proc. 40th IEEE Conference on Local Computer Networks (LCN)*. IEEE. 2015, pp. 462–465.
- [22] Nils Richerzhagen, Björn Richerzhagen, Rhaban Hark, Dominik Stingl, and Ralf Steinmetz. "Limiting the Footprint of Monitoring in Dynamic Scenarios through Multi-dimensional Offloading." In: *Proc. 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2016, pp. 1–9.
- [23] Nils Richerzhagen, Björn Richerzhagen, Michael Walter, Dominik Stingl, and Ralf Steinmetz. "Buddies, not Enemies: Fairness and Performance in Cellular Offloading." In: *Proc. 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. June 2016, pp. 1–9.
- [24] Nils Richerzhagen, Dominik Stingl, Björn Richerzhagen, Andreas Mauthe, and Ralf Steinmetz. "Adaptive Monitoring for Mobile Networks in Challenging Environments." In: *Proc. 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2015, pp. 1–8.
- [25] Julius Rückert, Björn Richerzhagen, Eduardo Lidanski, Ralf Steinmetz, and David Hausheer. *Optimizing Flow Topologies for Flash Crowd Support in Hybrid Peer-to-Peer Live Streaming*. Tech. rep. Peer-to-Peer Systems Engineering Lab, TU Darmstadt, Germany, 2014.
- [26] Julius Rückert, Björn Richerzhagen, Eduardo Lidanski, Ralf Steinmetz, and David Hausheer. "TopT: Supporting Flash Crowd Events in Hybrid Overlay-based Live Streaming." In: *Proc. 14th IFIP Networking Conference (Networking)*. IEEE. 2015, pp. 1–9.
- [27] Ralf Steinmetz, Melanie Holloway, Boris Koldehofe, Björn Richerzhagen, and Nils Richerzhagen. "Towards Future Internet Communications – Role of Scalable Adaptive Mechanisms." In: *ACADEMIA EUROPAEA* (2015), pp. 59–61.
- [28] Dominik Stingl, Reimond Retz, Björn Richerzhagen, Christian Groß, and Ralf Steinmetz. "Mobi-G: Gossip-based Monitoring in MANETs." In: *Proc. IEEE Network Operations and Management Symposium (NOMS)*. 2014, pp. 1–9.
- [29] Dominik Stingl, Björn Richerzhagen, Fabio Zöllner, Christian Gross, and Ralf Steinmetz. "PeerfactSim.KOM: Take it back to the Streets." In: *Proc. International Conference on High Performance Computing and Simulation (HPCS)*. IEEE. 2013, pp. 80–86.

- [30] Long Vu, Phuong Nguyen, Klara Nahrstedt, and Björn Richerzhagen. "Characterizing and Modeling People Movement from Mobile Phone Sensing Traces." In: *Pervasive and Mobile Computing* 17 (2015), pp. 220–235.
- [31] Markus Weckesser, Malte Lochau, Thomas Schnabel, Björn Richerzhagen, and Andy Schürr. "Mind the Gap! Automated Anomaly Detection for Potentially Unbounded Cardinality-based Feature Models." In: *Fundamental Approaches to Software Engineering*. Springer Berlin Heidelberg, 2016, pp. 158–175.

DEMO PAPERS

- [32] Christian Groß, Björn Richerzhagen, Dominik Stingl, Christoph Munker, David Hausheer, and Ralf Steinmetz. "Geodemlia: Persistent Storage and Reliable Search for Peer-to-Peer Location-based Services." In: *Proc. 13th IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE. 2013, pp. 1–2.
- [33] Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe, and Ralf Steinmetz. "Demo: Seamless Transitions Between Filter Schemes for Location-based Mobile Applications." In: *Demonstrations of the 41st IEEE Conference on Local Computer Networks (LCN-Demos)*. Nov. 2016, pp. 1–3.
- [34] Björn Richerzhagen, Marc Schiller, Max Lehn, Denis Lapiner, and Ralf Steinmetz. "Transition-enabled Event Dissemination for Pervasive Mobile Multiplayer Games." In: *Proc. 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2015.
- [35] Björn Richerzhagen, Stefan Wilk, Julius Rückert, Denny Stohr, and Wolfgang Effelsberg. "Transitions in Live Video Streaming Services." In: *Proc. Workshop on Design, Quality and Deployment of Adaptive Video Streaming (VideoNEXT)*. ACM. 2014, pp. 37–38.
- [36] Nils Richerzhagen, Björn Richerzhagen, Michal Lipinski, Markus Weckesser, Roland Kluge, Rhaban Hark, and Ralf Steinmetz. "Exploring Transitions in Mobile Network Monitoring in Highly Dynamic Environments." In: *Demonstrations of the 41st IEEE Conference on Local Computer Networks (LCN-Demos)*. Nov. 2016, pp. 1–3.
- [37] Stefan Wilk, Julius Rückert, Denny Stohr, Björn Richerzhagen, and Wolfgang Effelsberg. "Efficient Video Streaming through Seamless Transitions Between Unicast and Broadcast." In: *Proc. 2nd Conference on Networked Systems (NetSys)*. 2015, pp. 1–2.



CURRICULUM VITÆ

PERSONAL

Name	Björn Richerzhagen
Date of Birth	September 6, 1988
Place of Birth	Remscheid-Lennep, Nordrhein-Westfalen, Germany
Nationality	German

EDUCATION

Since 01/2013	Technische Universität Darmstadt, Doctoral candidate at the Department of Electrical Engineering and Information Technology
01/2011-11/2012	Technische Universität Darmstadt, Master studies in Electrical Engineering, Degree: Master of Science
10/2007-01/2011	Technische Universität Darmstadt, Bachelor studies in Electrical Engineering, Degree: Bachelor of Science
10/2006-09/2007	RWTH Aachen, Diploma studies in Electrical Engineering
08/1997-04/2006	Städtisches Röntgen-Gymnasium, Remscheid-Lennep, Nordrhein-Westfalen, Germany, Degree: Allgemeine Hochschulreife

ACADEMIC EXPERIENCE

Since 01/2013	Researcher funded by DFG (German Research Foundation) in the Collaborative Research Centre (CRC) 1053 – MAKI “Multi-Mechanismen-Adaption für das künftige Internet”
---------------	---

WORK EXPERIENCE

Since 01/2013	Technische Universität Darmstadt, Research assistant at the researcher group
---------------	---

Adaptive Overlay Communications at the
Multimedia Communications Lab (KOM)

TEACHING ACTIVITY

- | | |
|------------|--|
| 2013-2014 | Seminar “Mobile Phone-based Participatory Sensing and its Analysis in Mobile Communities”, Teaching Assistant. |
| 2013-2014 | Lecture “Algorithms for Mobile Networks”, Teaching Assistant. |
| since 2013 | Lab and Project “Multimedia Communications Lab/Project”, Organizer and Supervisor. |
| since 2013 | Seminar “Advanced Topics in Future Internet Research”, Organizer and Supervisor. |
| 2013 | Industrial Colloquium “Designing the Future Internet – Visions, Challenges, and Solutions”, Organizer and Speaker. |

HONORS

- | | |
|---------|--|
| 03/2015 | Award for “Efficient Video Streaming through Seamless Transitions between Unicast and Broadcast” at NetSys 2015. |
| 12/2014 | Best Demo Award for “Transitions in Live Video Streaming Services” at VideoNEXT 2014. |

Darmstadt, May 9th, 2017

Björn Richerzhagen

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, 9. Mai 2017

Björn Richerzhagen

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and \LyX : <https://bitbucket.org/amiede/classicthesis/>