TECHNISCHE
UNIVERSITÄT
DARMSTADT

QUALITY-AWARE TASKING in MOBILE OPPORTUNISTIC NETWORKS

Distributed Information Retrieval and Processing utilizing Opportunistic Heterogeneous Resources

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

von

THE AN BINH NGUYEN, M.SC.

Geboren am 24. März 1987 in Hanoi, Vietnam

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

Tag der Einreichung: 03. Juli 2018
Tag der Disputation: 30. August 2018

Hochschulkennziffer D17
Darmstadt 2018

The An Binh Nguyen, M.Sc.: *Quality-aware Tasking in Mobile Opportunistic Networks*
Distributed Information Retrieval and Processing utilizing Opportunistic Heterogeneous Resources.
Darmstadt, Technische Universität Darmstadt
Tag der mündlichen Prüfung: 30.08.2018
Jahr der Veröffentlichung der Dissertation auf TUprints: 2018

ABSTRACT

ADVANCES in wireless technology have facilitated direct communication among mobile devices in recent years, enabling *opportunistic networks*. Opportunistic networking among mobile devices is often utilized to offload and save cellular network traffic and to maintain communication in case of impaired communication infrastructure, such as in emergency situations. With a plethora of built-in capabilities, such as built-in sensors and the ability to perform even intensive operations, mobile devices in such networks can be used to provide distributed applications for other devices upon opportunistic contact. However, ensuring quality requirements for such type of distributed applications is still challenging due to uncontrolled mobility and resource constraints of devices. Addressing this problem, in this thesis, we propose a *tasking methodology*, which allows for assigning tasks to *capable* mobile devices, considering quality requirements. To this end, we tackle two fundamental types of tasks required in a distributed application, i.e., information retrieval and distributed processing.

Our first contribution is a decentralized tasking concept to obtain crowd collected data through built-in sensors of participating mobile devices. Based on the Named Data Networking paradigm, we propose a naming scheme to specify the quality requirements for crowd sensing tasks. With the proposed naming scheme, we design an adaptive self-organizing approach, in which the sensing tasks will be forwarded to the right devices, satisfying specified quality requirements for requested information.

In our second contribution, we develop a tasking model for distributed processing in opportunistic networks. We design a *task-oriented message template*, which enhances the definition of a complex processing task, which requires multiple processing stages to accomplish a predefined goal. Our *tasking* concept enables distributed coordination and an autonomous decision of participating device to counter uncertainty caused by the mobility of devices in the network. Based on this proposed model, we develop *computation handover* strategies among mobile devices for achieving quality requirements of the distributed processing.

Finally, as the third contribution and to enhance information retrieval, we integrate our proposed *tasking* concept for distributed processing into information retrieval. Thereby, the crowd-collected data can be processed by the devices during the forwarding process in the network. As a result, relevant information can be extracted from the crowd-collected data directly within the network without being offloaded to any remote computation entity. We show that the obtained information can be disseminated to the right information consumers, without over-utilizing the resource of participating devices in the network.

Overall, we demonstrate that our contributions comprise a *tasking* methodology for leveraging resources of participating devices to ensure quality requirement of applications built upon an opportunistic network.

KURZFASSUNG

Mit Hilfe von drahtlosen Kommunikationstechnologien ist der Aufbau von direkten (ad hoc) Verbindungen zwischen Mobilgeräten möglich. Über diese Verbindung können Informationen lokal und direkt unter Mobilgeräten ausgetauscht werden, was eine Grundlagen für viele Anwendungen bietet. Der sinnvolle Einsatz solcher Technologien ist zum Beispiel die Entlastung der Mobilfunknetze durch lokale Datenverteilung und der Aufbau sowie die Aufrechterhaltung von infrastrukturlosen Netzen in Katastrophenszenarien. Darüber hinaus verfügen Mobilgeräte über eine Vielzahl an leistungfähigen Sensoren, und die Möglichkeit, komplexe Aufgaben auszuführen. Solche Fähigkeiten können als Dienste im Kontext einer verteilten Anwendung mit direkter Kommunikation angeboten werden. Dabei ist eine große Herausforderung, die Qualitätanforderungen solcher Anwendungen bei limitierten Ressourcen und der Mobilität der Geräte zu garantieren. In dieser Arbeit wird diese Problematik addressiert, indem wir ein Modell zur verteilten Aufgabenzuweisung unter teilnehmenden Mobilgeräten entwickeln. Bei dieser Verteilung werden vor allem unterschiedliche Qualitätsanforderungen und die Heterogenität der Geräte berücksichtigt. Dabei fokussieren wir uns auf die Informationsakquise und die verteilte Datenverarbeitung, welche die Grundsteine für die Entwicklung einer verteilten Anwendung darstellen.

In unserem ersten Beitrag entwickeln wir eine Methode basierend auf dem *Named Data Networking* Paradigma, um die Qualitätanforderung einer Anfrage zum Datensammeln zu spezifizieren. Dabei werden die Anforderungen direkt in das Namensschema integriert. Aufbauend auf dieser Spezifikation wird die Weiterleitung von Anfragen an Daten durch Geräte im Netzwerk selbst organisiert und gestaltet, sodass die Anforderungen an die angefragten Daten erfüllt werden können.

Mit unserem zweiten Beitrag wird eine verteilte und lokale Verarbeitung von Daten auf Basis der Kooperation mit anderen Mobilgeräten untereinander ermöglicht. Um eine Verteilung der Verarbeitung auf benachbarte Mobilgeräte zu ermöglichen, wurde eine Nachrichtenvorlage konzipiert, welche ein komplexes Verarbeitungsziel spezifiziert, in mehrere Operationen aufbricht und an umliegende Mobilgeräte verteilt. Dabei erlauben wir autonome Entscheidungen der teilnehmenden Geräte, um der hohen Dynamik und der Ressourceneinschränkung entgegen zu wirken.

Als letzten Beitrag kombinieren wir das entwickelte Model zur verteilten Datenverarbeitung mit unserer Methode zur Weiterleitung von Netzwerkpaketen. Dabei werden die angefragten Daten während der Weiterleitung verarbeitet, um die relevanten Informationen direkt im Netzwerk zu extrahieren. Wir zeigen, dass solche Informationen effizient interessierten Mobilgeräte geliefert werden können, ohne dabei zusätzliche Last in Netzwerk zu generieren.

Die in dieser Arbeit vorgestellten Beiträge bilden gemeinsam ein Model zur Aufgabenzuweisung, welches die Qualitätanforderungen einer Anwendung in dynamischen Mobilnetzen erfüllt.

ACKNOWLEDGEMENTS

As commonly known, pursuing a Ph.D. is a tough adventure. This adventure consists of many hardships and burdens that one could hardly overcome just by himself. Now that my Ph.D. adventure has finally come to an end, I want to look back the road that I have been through, and want to express my sincere thanks to the people, who have supported and accompanied me through this steep road.

First and foremost, I want to thank my professors. Thank you, Ralf Steinmetz, for giving me the chance to work, conduct research, and finish my Ph.D. at KOM. You are a fantastic *Chef*, always know how to best support your lab members. Also, thank you for being a very kind professor towards the students. I was once your student in a lecture, and your being a kind professor was definitely one significant factor, that motivated me to start my Ph.D. with you. Furthermore, I want to express my gratitude to my mentor, Michael Zink. Thank you for your insightful feedback for my papers and my dissertation. Your relaxed nature was always an inspiration for me.

I have faced several dead-ends during the course of my Ph.D.; for that, I could not have survived without the support of the awesome people of *DSS* group. Thanks to my group leaders, in historical order, Sonja, Christian, Doreen, and Björn. You have helped me a lot by offering your guidance. A big thank you to my office mates, the first generation Frank and the second generation Tobi (Meuser). You were my amazing office mates, always there for me, hearing me out and comforting me when I was in difficult situations. Also a big thank you to Patrick, my comrades-in-arms in the *disaster relief* projects. I would like to thank Tobi (Rückelt). You were a really nice colleague, always willing to help me or to just simply chat with me at any time. Of course, I cannot forget the other members of the DSS group, Alaa, Daniel, and Florian. Thank you to all DSS members.

Besides the DSS group, it was my pleasure and honor to meet other amazing people at KOM. I want to express my gratitude to the following KOM members especially. Thank you to Jeremias for being my *eating buddy* and for becoming my friend. Thank you to Lena for many of our conversations. Thank you to Karola for being the KOM grandma. Thank you to Thomas and Robert for being members of the Pizza team. Thank you to Ms. Scholz-Schmidt, Ms. Ehlhardt, Sabine etc. for making our life easier by supporting us in administrative matters. Thank you to Alaa, Wael, and Viktor for your teamwork in supervising the KN1 lecture. Thank you to the whole KOM lab, for being a great, big family for me. I will always remember *ALL* of you.

I would like to express my thank to the colleagues in the NICER and SMARTER projects. You are all talented and great researchers. Especially, many thanks to Chris for your close collaboration with me in the NICER project. Many contributions in my dissertation were proposed thanks to our discussion.

I had the pleasure of working with several incredible students. I want to thank Pratyush, Christian, and Long for your support.

Finally, and most importantly, I want to thank my family. My late father and my mother have sacrificed a lot for me. My wife, Hang, has been going through many difficulties with me, and will accompany me with many years to come. Thank you for being in my life. I love you all.

CONTENTS

1 INTRODUCTION

In a paper from early 1997, Satyanarayanan [170] highlighted the potential of mobile computing for the next decade. Several application scenarios are envisioned, such as mobile health-care which utilizes mobile devices for remote consultation and emergency response scenarios which mostly require information obtained from participating devices for better coordinating the relief works. A common ground of the envisioned application scenarios is the opportunistic utilization of resources available on mobile devices. This observation can be confirmed, looking into the development of mobile devices related research topics in the last few years.

Today, a mobile device is capable of several tasks; for instance, obtaining information either through active user's inputs or leveraging the built-in sensors to sense the environment opportunistically. The early adoption of this paradigm was *people centric sensing* [26], which focused on environmental data collection, actively triggered by the users. By combining the active user's input with *opportunistic sensing* through built-in sensors, this sensing paradigm was extended to the more general paradigm of *crowd sensing* [63]. Compared to the classical Wireless Sensor Network (WSN), which relies on pre-deployed static sensors to collect data, *crowd sensing*, which leverages mobile devices to collect information, provides a more flexible framework to collect information.

In addition to the ability to obtain information in various forms, modern mobile devices are also capable of processing information and executing complex operations, e.g., image processing [84] and data analysis using machine learning techniques such as TensorFlow[1]. Furthermore, under the proliferation of mobile devices in the last few years, offloading intensive computation to a set of mobile devices in an opportunistic network emerges as a new, promising solution for distributed processing [207]. Compared to the legacy computation offloading to a remote cloud for processing, opportunistic offloading provides several advantages: (i) reducing deployment cost, since the idle resources of nearby mobile devices will be utilized [223], (ii) reducing network traffic flowing through the communication infrastructure [76], and (iii) potentially reducing responses delay [122], since the mobile devices can communicate with each other directly through common available wireless communication, such as WiFi-direct, or Bluetooth.

Communicating directly with each other through wireless technologies such as ad hoc WiFi, recently WiFi-direct, or Bluetooth is the fundamental enabler for the utilization of mobile devices in the aforementioned application scenarios. Early research on Mobile Ad Hoc Network (MANET) allows us to construct and maintain communication among nodes in an ad hoc network; however, in consonance with the development and proliferation of mobile hand-held devices, there is a shift towards people-centric

1 https://www.tensorflow.org/mobile/

networking [34]. The main driver behind people-centric networking is the mobility of the human carriers. On the one hand, human mobility makes the network more dynamic and unstable; on the other hand, mobility can also be used to deliver message to before unreachable network partitions, which is the main idea behind mobile opportunistic networking [95]. The usage and deployment of mobile opportunistic ad hoc network in practice have been studied and proved to be credible; for instance, the BMBF funded project SMARTER[2] enables information exchange in disaster relief situations through opportunistic ad hoc network, formed by mobile hand-held devices.

Despite the plethora of advantages, which are available through the utilization of mobile devices, building services and applications over an opportunistic network formed by such devices still has not found the widely acceptance it deserves; as stated by Lindgren et al. [99] "What would be the Killer app for opportunistic networks". This question arises mainly due to the dynamic nature of opportunistic network, formed by heterogeneous mobile devices, which makes it a challenging task to ensure the quality of applications and services on top of such network.

Focusing on this challenge, in this thesis, we propose a *tasking* methodology, which addresses the question of how to ensure quality for services and applications built upon opportunistic networks.

1.1 MOTIVATION FOR OPPORTUNISTIC RESOURCE UTILIZATION

Opportunistic resource utilization is first introduced as a paradigm, generally focusing on application built upon specialized ad hoc network, such as opportunistic network [96]. Thus, opportunistic resource utilization is a well-suited model for scenarios, which mainly relies on opportunistic ad hoc network as the main communication medium, such as in emergency response scenarios, in which the communication infrastructure might be impaired or inaccessible [93]. In the work at hand, we focus on providing Quality of Service (QoS) for applications and services, that are based on *opportunistic resource utilization*.

Since mobility is the main driver for an opportunistic network, and an end-to-end path between devices in an opportunistic network is not always possible, a plethora of forwarding mechanisms have been proposed, which mainly focus on increasing the messages delivery rate [17, 24, 92, 98, 147]. Some applications, directly profiting from high delivery rate of opportunistic forwarding are, for instance, information dissemination, e.g., notification in emergency response [115], or traffic information in vehicular networks [145]. However, increasing messages delivery rate in opportunistic network in general does not automatically ensure the quality requirements of applications and services.

Following the paradigm of *opportunistic resource utilization*, Conti et al. introduce and discuss the advantages of the so-called opportunistic computing [35]. Here, a device upon opportunistic contact with other devices or infrastructure, can leverage the idle, available computing resource of these to offload computation. Many approaches

2 http://smarter-projekt.de/

for opportunistic offloading to date offload computation to one-hop edge computing infrastructures [114, 119, 122, 216], with two main targets: i) saving energy of devices with low computing-capacity, and ii) reducing network traffic for communication infrastructure required for computation offloading to a remote cloud server. Accordingly QoS is often measured by the amount of energy for the offloading devices, or how much network traffic can be saved. With regards to offloading over multiple hops, primarily targeting mobile devices, several approaches dealing with services composition as a type of computation offloading in opportunistic network have been proposed [146, 150, 166, 187]. The main QoS targets of services composition in opportunistic network lie in optimizing response time, and success rate of the services execution. Another important target for opportunistic offloading is load balancing [167], benefiting overall performance, and increasing user's acceptance. While the aforementioned approaches deal with different aspects of computation offloading for mobile devices, they still lack support for adaptation and for enabling autonomous decisions of the participating devices, which is required due to the dynamic nature of opportunistic networks.

Despite the flexibility, acquiring distributed information leveraging built-in sensors and user's input through mobile devices, i.e, *mobile crowd sensing*, introduces another dimension of quality requirements, i.e., quality of information [62]. With regards to information, quality requirements can be simple, such as which sensor to use, when to trigger data collection etc. [215]. The quality requirements can also be complex, for instance to cover up the sensing area [77, 206]. Most of the approaches, however, rely on a centralized coordination entity to keep track of the participating mobile devices, in order to formulate an optimization problem and accordingly derive a solution for sensing tasks allocation. Several distributed recruitment frameworks [75, 195] attempt to realize crowd sensing on opportunistic networks, these, however consider the coverage problem of a sensing area, therefore assume a rather homogeneous setup.

In this thesis, we explicitly consider the heterogeneity to assign *tasks* to the capable devices, aiming to leverage their resources and capabilities opportunistically to ensure quality requirements.

1.2 RESEARCH CHALLENGES

While providing QoS is a non-trivial task in centralized systems, it becomes even more challenging in opportunistic networks to satisfy quality requirements due to the mobility and rapid changes of participating devices. With respect to the goal of the thesis as aforementioned and to the focus on mobile opportunistic networks, the following research challenges are identified:

Challenge: *Rapid changing context of the participating devices, resulting from mobility, heterogeneity and resource constraints.*

The two most important aspects characterizing the rapidly changed context of the devices, that need to be considered for tasking in a mobile opportunistic network are the mobility and the heterogeneity. The mobility and the heterogeneity of the participating devices can be considered as both chances, as well as challenges for tasking in mobile opportunistic networks. On the one hand, the mobility makes the network unstable, leading to rapidly changed network topology. On the other hand, the mobility of of the devices can be utilized to bridge the communication in case the end-to-end path among devices are not available at particular time. Similarly, the heterogeneity of the participating devices can be leveraged in a collaborative processing of complex task. To this end, a complex task can be divided in several simple tasks, which require a special operation, a particular amount of resources, available due to the heterogeneity of devices in the network. Nevertheless, the heterogeneity together with the mobility of the participating devices can also result into the unavailability of required resources at particular time. All in all, how and when to leverage, or to countermeasure mobility and heterogeneity have to be specially taken into consideration. Last but not least, the participating devices in a mobile opportunistic network are mostly resource constraint w.r.t. computation capacity, energy level etc., which adds up to the complexity of the task allocation.

Challenge: *Decentralized and distributed coordination.*

As discussed in the motivation, access to a centralized coordination entity cannot always be guaranteed for the participating devices. Furthermore, the devices in a mobile opportunistic network tend to move and change their location frequently, which makes a centralized coordination both remote as well as in a close proximity, e.g., through clustered-head devices, less feasible. Consequently, the devices in a mobile opportunistic network should coordinate with each other in a decentralized distributed manner. Due to the fact, that each device in the opportunistic network only possesses a partial view of the network and due to rapidly changed context as elaborated in the first challenge, distributed and decentralized coordination can lead to inefficient resources utilization and generate much overhead. Thus, achieving a good performance, while generating less overhead through distributed coordination is the second challenge that needs to be addressed.

1.3 RESEARCH GOALS AND CONTRIBUTIONS

The main objective of this work is to develop *a tasking methodology for mobile opportunistic networks*, providing mechanisms to utilize the available resources and capabilities of the participating devices in such opportunistic network to (i) *acquire* information, (ii) *process* the acquired information to extract more valuable, situation-aware information, and (iii) *disseminate* information to the information-consumers efficiently, while ensuring quality requirements. Accordingly, the main objective is translated into three major research goals

Research Goal 1: *Decentralized mechanism to distribute sensing tasks.*

To acquire information, the built-in sensors in mobile devices can be leveraged. However, not all information and sensing data are relevant. Particularly, relevant data requested by an information consumer should satisfy spatio-temporal requirements as being specified by this information consumer. To this end and to account for the aforementioned challenges, we contribute a naming scheme and a forwarding protocol to allow for efficient sensing tasks distribution in decentralized opportunistic network [131]. The proposed naming scheme incorporates the tasks requirements, which allow the participating devices to act autonomously, thus facilitating decentralized and distributed coordination. The forwarding mechanism is designed aiming to leverage mobility of the participating devices to forward the sensing tasks to the capable mobile producers.

Research Goal 2: *Mechanism to enable distributed in-network processing.*

The information as being acquired through the contribution stated in research goal 1, has to be further processed to extract more valuable and high-level information. Since processing information might require several (complex) operations, a local processing on a single device might consume much energy, rendering resource-constraint devices useless. Therefore, the processing can be offloaded to a remote cloud or to devices in close proximity for collaborative computation. While the former is often used in general, an access to a remote cloud server is not always guaranteed (impaired communication infrastructure, urban rural areal with limited access, failure and unavailable cloud server). Hence, we focus on the later, i.e., collaborative processing of information utilizing heterogeneous resources of devices in opportunistic networks. To this end, we propose a mechanism to enable distributed processing of complex processing tasks and to facilitate autonomous decision of the participating devices aiming towards decentralized, distributed coordination [134]. Accordingly, we design computation handover strategies for efficient distributed processing [136].

Research Goal 3: *Mechanism to deliver results to mobile information consumers.*

The results of crowd sensing tasks and distributed processing need to be delivered to the right consumers. While information dissemination is intended for multiple information consumers, results delivery is more targeted. Thereby, common dissemination mechanisms in opportunistic networks, which mostly rely on flooding and replicating messages are inefficient for results delivery. Furthermore, results delivery becomes more complicated if the information consumers are also mobile. We address this problem in our third and last research goal. We design a mechanism to integrate mobility prediction to estimate the future locations of mobile consumers for results delivery. Thereby, results delivery based on mobility prediction needs to be accurate, achieve low latency while generate less overhead.

In order to realize resources utilization in an opportunistic network, one has to assume cooperative behavior of the participating devices. Thus, trust among the devices is desired. We assume that the devices can set up a trusted environment among each other in a distributed manner such as [44], and do not provide our own mechanism targeting trust. Additionally, privacy is also an important aspect, since we rely on distributed coordination and on context information shared by participating devices to enable autonomous decision making. Considering the privacy for participating devices, the information disclosed in the shared context information in our mechanisms only reveals sufficient information required for either forwarding or processing tasks. Overall, we assume that the participating devices cooperate with each other in a collaborative manner and do not have any malicious intentions.

1.4 STRUCTURE OF THE THESIS

In this chapter, we have elaborated on our main research goal to develop a *tasking methodology*, that leverages the resources of mobile devices to ensure quality requirements for applications and services built upon opportunistic network. Following the introduction, in Chapter 2 we provide an overview of background information with regards to the enabling technologies to achieve the contributions of this thesis. Based on the background, we review and discuss the related work, focusing on three main aspects of our contributions, namely, (i) location based forwarding in opportunistic networks as well as interest forwarding in Named Data based mobile networks, (ii) distributed processing, and (iii) information dissemination.

The contributions of this thesis constitute an *information retrieval workflow*, that involves three main steps (i) creation and assignment of information (sensing) tasks, (ii) distributed processing of crowd collected data within networks, and (iii) delivery of the processed data to the right *information consumers*. According to these steps, we will present the contributions in Chapter 3, 4, and 5. We propose the mechanism for crowd sensing tasks distribution as the first step of *information retrieval* in Chapter 3, addressing the mobility and the heterogeneity of the *information producers*. For the second step of *information retrieval*, to process the collected data directly in an opportunistic network, we leverage the idle resources available on participating devices. We introduce our distributed processing mechanism focusing on opportunistic networks in Chapter 4. To wrap up the *information retrieval workflow*, we present the concept for *results delivery*, addressing the mobility of *information consumers* in Chapter 5.

Finally, the thesis will be concluded in Chapter 6, summarizing our core contributions. Thereby, we provide a discussion on potential research directions taking the contributions of this thesis as a basis.

2 BACKGROUND AND RELATED WORK

In this chapter, we provide background information, discuss the enabling technologies, and review the state of the art which are relevant for our research goals. First, in Section 2.1, we elaborate on several sensing paradigms as the basis for information retrieval. We consider emergency response as the motivating scenario for our contributions. Given the impaired communication infrastructure in emergency situation, we rely on forwarding in an opportunistic network as communication medium to provide services and applications. Therefore, in Section 2.2, we review related work for message forwarding in opportunistic networks. Particularly, for the purpose of crowd sensing tasks distribution, location-based forwarding concepts such as geocast in opportunistic networks are of interest. Furthermore, the *information request* forwarding concepts to retrieve information in Named Data Networking (NDN) in mobile networks, that are primarily built upon wireless direct ad hoc communication such as MANET and Vehicular Ad Hoc Network (VANET), are closely related to our work. Second, in Section 2.4, we analyze related work for distributed data processing, in which the processing functions are distributed and carried out by the nodes directly in the network, without offloading to any cloud server. Thereby, we review in-network processing concepts of relevant research contexts such as WSN, Complex Event Processing (CEP) operator placement, and enabling techniques for distributed processing in opportunistic network, such as Delay-tolerant Networking (DTN) based Remote Procedure Call (RPC), Named Function Networking (NFN). Third, for information/results delivery, forwarding concept of mobile opportunistic network and NDN based mobile network can also be leveraged. Thereby, we concentrate on the state of the art for information/results dissemination in opportunistic network, as well as *data forwarding* for NDN based networks. We conclude the chapter by discussing the challenges for providing QoS for applications and services built upon opportunistic network and consequently investigating the research gap.

2.1 INFORMATION RETRIEVAL THROUGH SENSING PARADIGMS

Information retrieval is a term with a very broad meaning. For further discussion, we refer to the definition coined by Larson from an academic perspective with regards to computer systems as follows:

Definition 2.1: Information Retrieval

"Information retrieval is finding material [...] that satisfies an information need from within large collections (usually stored on computers)." [86]

As per Definition 2.1, three aspects of information retrieval are apparent (i) *information need*, i.e., requirements that have to be defined for the purpose of getting information, (ii) *finding material*, i.e., data in several forms that can be used to extract information, and (iii) *large collections*, which refers to the scale and distributed manner of information location. *Information retrieval* can occur, for instance, when a user searches on his computer, looks for something in the Internet, or collects data from a sensor network. With respect to the context of this thesis, we refer to the act of *information retrieval*, as acquiring information from geographically distributed sensors of different types. In the following, we discuss the two most important sensing paradigms, i.e., *wireless sensor network* and *crowd sensing*.

Wireless Sensor Networks — WSNs: WSN technology has gained attention, both in academic research and production systems due to its great potential for application [12, 42]. Information retrieval with WSN technology relies on sensors, which are in general small devices with capabilities to capture data from the surrounding environment, e.g., temperature, humidity etc.; sensors are also capable of networking via wireless communication standards such as ZigBee/IEEE 802.15.4 [69]. Furthermore, a sensor also possesses a processing unit, which allows for simple data processing. Multiple sensors can form an ad hoc network and allow the user to interact with the WSN network. Within the WSN network, several nodes can be elected as *sink* nodes, which on the one hand serves as an interface that receives user's command, and on the other hand receives the data collected from the sensor network. Through the *sinks* as interfaces, a user is able to inject sensing or processing tasks to the sensor network, which will be forwarded to the appropriate sensors for execution. Due to the fact, that sensors are small low-cost devices, mostly powered by a battery, energy efficiency is the biggest challenge to tackle in WSNs. A plethora of research work has been proposed to address the energy issue of WSN from different perspective, e.g., Medium Access Control (MAC) techniques [22], routing techniques for WSN [205], topology control techniques [169], and data processing, e.g. filter, aggregation within sensors networks [208]. Overall, the concepts and techniques developed for retrieving information from WSNs provide a foundation, which inspires other sensing paradigms as well as information retrieval systems. In comparison to opportunistic networks (cf. Section 2.2), a WSN is typically of small scale, and does not fully consider rapid changes and mobility of participating devices.

To create an application using WSN technology, the sensors have to be deployed in advance, which induces deployment cost and reduces the flexibility of the technology. Recently, inspired by WSN the *crowd sensing* paradigm emerged with more flexibility with regards to deployment.

Crowd Sensing: Due to the proliferation of mobile devices in recent years, and to the fact that modern mobile devices are equipped with diverse built-in sensors, the crowd sensing paradigm has emerged as a new, flexible model to acquire distributed information. Crowd sensing can take on two forms, i.e., *participatory sensing* and *opportunistic sensing* [85]. While in participatory sensing, the requested information is collected upon being actively triggered by the device's carrier; in opportunistic sensing, the requested information is collected when certain conditions are met, without requiring any interaction from the human carrier. For this thesis, we abstract from both crowd sensing forms, and assume that if a device receives a sensing task, the task will be executed. Figure 1 shows the illustration for a common crowd sensing system architecture.

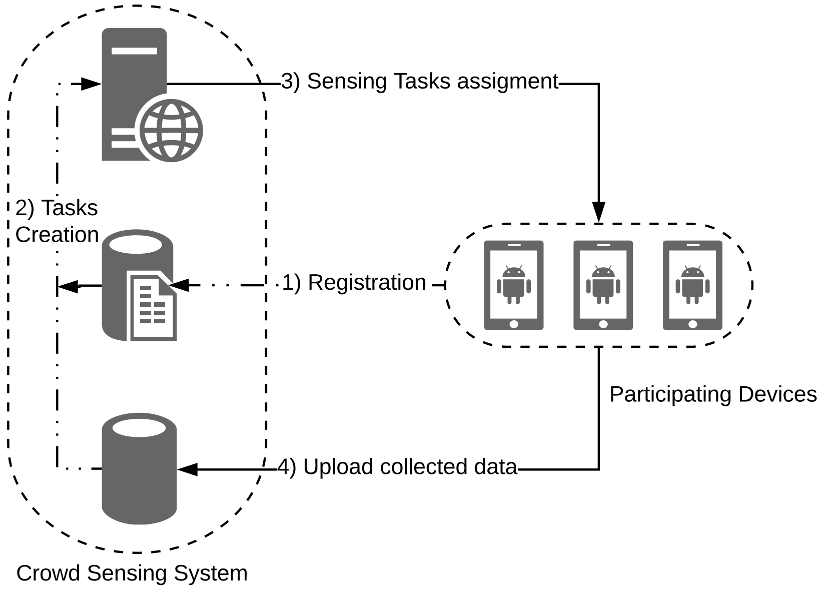


Figure 1: Illustration of a general architecture for crowd sensing systems (adapted from [151]).

A typical crowd sensing system consists of two parts; one part is implemented in form of distributed servers for management purpose, and another part is implemented on the participating devices in form of an application [55]. The participating devices can register themselves with the management server, to inform the server on its willingness to receive and execute the appropriate sensing tasks to its capabilities (step 1). A user, who wants to retrieve particular information, can interact with the crowd sensing system, to create sensing tasks based on the information need (step 2). Thereafter, the crowd sensing system relies on a list of available devices obtained through the registration, to assign the sensing tasks to the devices, that can execute the tasks (step 3). To facilitate sensing tasks creation (step 2) and tasks assignments/execution (step 3), a machine readable abstraction/representation might be required. For instance, Medusa [155] provides such abstraction, and a corresponding semantic to specify the goal of the sensing task, the steps to execute the sensing tasks on the participating devices, which can be interpreted and scheduled for execution by the application running on the participating devices. Finally, the data collected from the participating devices will be uploaded to the server, utilizing communication interfaces available on the participating devices (step 4). Since a mobile device possesses multiple communication interfaces, several upload options are possible, e.g., directly to the server via LTE, WiFi network when connected, or indirectly through multiple-hop ad hoc communication to forward data to some gateway devices [111].

Compared to WSN, crowd sensing provides more flexibility and reduces deployment cost, since crowd sensing utilizes existing mobile devices as sensors. Consequently, crowd sensing introduces several inherent characteristics [55]: (i) crowd sensing enables data collection from a large scale scenario, given the participation of a sufficient number of devices to cover the area, which is not possible with preinstalled sensors, (ii) as a tradeoff, the quality of data collection can vary due to heterogeneous configuration and quality of sensors available on participating mobile devices, (iii) while WSN in-network processing units are only capable of simple data processing, mobile devices are capable of executing complex operations; consequently the collected data can be processed and analyzed completely within the network without offloading to a remote computation entity. However, even though mobile devices are capable devices, they still suffer from resource constraints. A crowd sensing application might have to share the resources with other applications running in the mobile devices and a crowd sensing application might have to serve several sensing tasks in parallel.

Research on crowd sensing has focused on several aspects. Many works deal with practical aspects of crowd sensing with regards to definition and implementation of crowd sensing applications, such as automotive-oriented applications [51, 143], healthcare [152, 153], disaster relief [108, 212], and design of middleware solution to develop crowd sensing applications on participating devices [2, 155, 179, 213]. Other research directions of crowd sensing, which focus more on the theoretical perspective, are to recruit more devices to participate in the crowd sensing application, and to assign sensing tasks to satisfy predefined requirements, taking into consideration the highly dynamic nature of mobile devices [157]. To date, the most commonly used quantification for crowd sensing quality used in research is still sensing area coverage and cost minimization [62]. Accordingly, the most common approaches rely on the formulation of an optimization problem, taking into account the current state of available participating devices, such as current energy level [72, 100], the total budget for the crowd sensing campaign [102, 180], as constraints to determine a tasks assignment, satisfying both the constraints and the area coverage. Thereby, these approaches have to assume that the participating devices can maintain a connection to the tasking servers of the crowd sensing system, in order to track and send requests to these devices. However, a persistent connection between participating devices and the tasking servers cannot always be guaranteed, e.g., in emergency response scenarios. Ma et al. [111] point out, that the WiFi interface of mobile devices can be utilized to enable communication directly among participating devices via opportunistic contacts, allowing for sensing tasks assignment and resource utilization even if the central tasking servers are not reachable, consequently ensuring the successful information retrieval. This type of communication belongs to the *opportunistic networks* paradigm, which will be discussed next.

2.2 opportunistic networks

Opportunistic networks are considered as a subclass of Delay-tolerant Networking (DTN). Therefore, we first briefly review the concept of DTN. DTN-based communication [48, 49] is proposed with focus on *intermittently connected networks*, in which the connections between network entities are only short-lived. The biggest challenge in intermittently connected networks is that the end-to-end connection between network entities cannot always be guaranteed. As a consequence, flow based Internet protocols such as TCP/IP cannot function properly in such partitioned networks. Examples of *intermittently connected networks* can be found in interplanetary networks for deep space communication [5], or in disaster situations, in which the communication infrastructure might be impaired and the network becomes divided into several regions. To allow for communication and for inter-operability among separated regions of the network, DTN relies on an overlay protocol, named *bundle* layer.

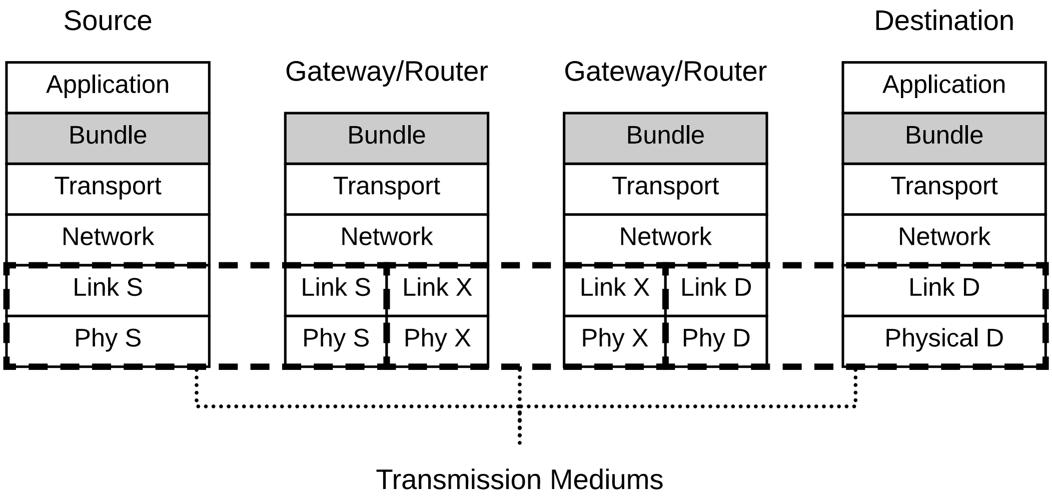


Figure 2: Illustration of the protocol stack with bundle in DTN communication (adapted from [79]).

The bundle layer [173] resides between application layer and transport layer as shown in Figure 2. A *bundle* encapsulates all data and control information within a single entity, which can be delivered asynchronously from source to destination. Each bundle can travel through several gateways or routers. It is assumed, that the pairwise transmission of a bundle between two DTN network entities is reliable. Thereby, the transmission mediums between each pairs of DTN entities can be different, e.g., Ethernet, wifi, satellite based communication. Consequently, the inter-operability among different network regions can be realized. Each DTN gateway or router is capable of storing a bundle, therefore, enables asynchronous delivery of a bundle through *store and forward*, in which each network entity first stores the bundle, and later forwards or delivers the bundle when connected to other network entities or to the destination. The concept of DTNs and bundle as transmission entity can be generalized with a bundle being a message. *Mobile opportunistic networking* paradigm realizes this generalization as a subclass of DTN concept. Mobile opportunistic networking, which can also be referred to as only opportunistic networking, focuses on asynchronous delivery of messages. The illustration of an opportunistic network is shown in Figure 3.

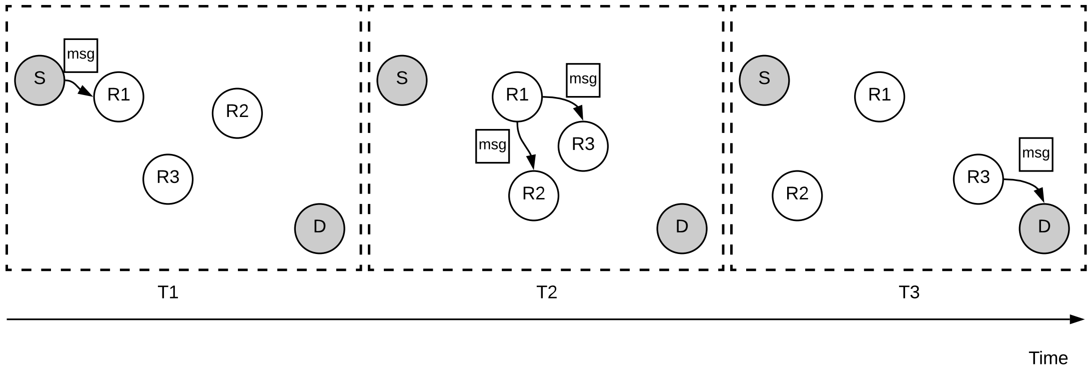


Figure 3: Illustration of opportunistic forwarding to deliver a message.

Since an end-to-end path between sources and destinations in an opportunistic network cannot always be guaranteed, we refer to the message delivery process as *opportunistic forwarding*, since the term *routing* is normally used when a route can be established between source and destination, e.g., in MANET [188]. The objective of the forwarding process in opportunistic networks is to successfully deliver messages to the intended destination, while minimizing the resources used by the participating devices [175]. Opportunistic networking focuses primarily on devices which are highly mobile and can communicate with each other through a wireless communication interface upon opportunistic contact. Each opportunistic contact can be used as a chance to forward or to replicate a message. Due to the fact that in opportunistic networks the devices are highly mobile, mobility of devices can be leveraged for message delivery. Thereby, opportunistic networks extend the *store and forward* concept of DTN to the *store, carry, and forward* principle to deliver messages. After receiving a message, a device can store the message, carry the message, thus becomes a *message ferry* and later forward either to other devices with better chance to reach the destination, or deliver to the destination itself. As mentioned previously, opportunistic network is a generic networking paradigm. The concept can be implemented upon wireless communication interfaces of mobile devices in different ways. For instance, Doering et al. [43] base the implementation to allow for opportunistic networking among Android based devices on the original specification of the bundle overlay [43]. This implementation can work for ad hoc WiFi, WiFi direct, as well as Bluetooth. Baumgärtner et al. [13] extend Serval [57], a middleware developed for providing services on wireless mesh ad hoc networks for mobile devices based on the 802.11 standard, with *store, carry and forward* capabilities, thus realize opportunistic network. In general, any self-encapsulated message entities transported pairwise over wireless communication interfaces among mobile devices, are compatible for being used in opportunistic networks.

2.2.1 *Forwarding Mechanisms*

In this section, we first briefly review several prominent common forwarding approaches for opportunistic networking, before going into more details of related work for location-based forwarding and interest forwarding.

First to be mentioned is *epidemic forwarding* [200] proposed as a reliable solution for successful message delivery in opportunistic networks. Reliable message delivery is achieved by letting devices exchange and synchronize the messages in their buffers upon each opportunistic contact, i.e., flooding the whole network. Thereby, all participating devices will receive all messages sooner or later. It is clear, that this approach has a trade-off between redundancy and reliability. Following up on the redundancy draw-back of epidemic forwarding, several forwarding approaches attempt to reduce the number of redundant messages, being replicated in the network, while still aiming to achieve successful delivery of messages in a timely manner. One prominent example of approaches relying on message replication is spray and wait [181], which injects several copies of a message into the network, and wait for these to be successfully delivered. An improved version of spray and wait is proposed, i.e., spray and focus [182] which instead of waiting in the second phase, tries to forward messages to devices with higher probability the reach the destinations. The probability for the focus phase in spray and focus is calculated based on the last encounter time with the destination. Several approaches dig deeper into the direction of estimation and prediction to further regulate the forwarding decision. For instance, PROPHET [98] bases the forwarding decision on an aging metric, which lets each device estimate the probability that it can meet other devices in the network. In PROPHET, more frequent meetings mean a higher chance one device can receive messages intended for the other; however, less frequent meetings will result in a decay value, which consequently decreases the chance of a device to become a relay node. Lately, with the shift towards more people-centric opportunistic networks, many forwarding approaches thus leverage context information of devices, which in turn includes the context information of their human carriers. Under this category, we mention prominent approaches such as HiBOP [17], profile-cast [70, 71], and CAR [130]. The common idea behind forwarding approaches, that use context information for message deliveries, is to create a profile of participating devices based on the context information and to match the profile of the relay devices with the destination, in order to derive the similarity. Devices with higher value of the similarity metrics will be favored during the forwarding process. Context information used in these approaches can be social-based metrics, such as community, group, frequent visiting locations etc.

The general forwarding approaches for opportunistic networks focus on the delivery of messages to a specified destination. In recent years, location-based services, e.g., location-based data collection in emergency response [105], is gaining importance. Consequently, for location-based services built upon opportunistic network, messages are not only intended and delivered to specified recipients, but also for region, for a group of recipients, which fulfill location requirements. Accordingly, next we review and discuss the related work for location-based forwarding in opportunistic network, as well as several location-based applications/services built on this type of network in detail.

2.2.2 *Location-based Forwarding*

Location-based forwarding in opportunistic networks can be considered from two perspectives, infrastructure-supported and infrastructure-less communication architecture. For several circumstances, such as in an emergency response scenario, a hybrid communication infrastructure might still be available, such that part of the network relies on communication infrastructure, while part of the network has to rely on a flat opportunistic ad hoc network. Thereby, several gateway devices can be selected to connect partitioned network regions [39]. Location-based forwarding thus can be realized by forwarding to gateway devices near the designated regions [117]. This approach is also quite common in VANET, in which Road Size Units (RSU) and road segments can be leveraged for location-based forwarding [90, 97, 125]. However, in the considered emergency response scenario, the coverage of such gateway devices over the designated region cannot always be guaranteed, accordingly, very often in the last mile of the location-based forwarding, the devices have to rely on an infrastructure-less opportunistic ad hoc networks. Next, we review and discuss selected location-based forwarding approaches designed for opportunistic networks.

Obviously, *epidemic forwarding* can also be used for location-based applications in opportunistic networks. However, due to the high overhead caused by flooding, this approach is often used as a baseline for comparison. Location-based forwarding mainly relies on context information such as location, movement speed, moving directions of the devices. Thereby, location-based forwarding has to assume that each device is capable of determining its current location, and movement; thus each device is capable of estimating its current distance to the designated geographical regions.

Using distance information from participating devices has been proposed since early research in location-based routing designed for MANET, with the location-based aided routing (LAR) [80]. LAR defines the *expected zone* where the destination might appear, and the *request zone*, covering the *expected zone*, which serves as the designated region for the routing decision. In LAR approach, distances of the MANET nodes are broadcast towards the source, so that this can build corresponding routes, leading towards the direction *request zone*, by choosing nodes with minimum distance for the next hop. For opportunistic networks, since an end-to-end path does not always exist, consequently, the forwarding decision is made by the mobile devices along the way.

The MOVE framework introduced in [87] is one of the first location-based forwarding schemes designed for opportunistic networks. In the MOVE approach, each node estimates the *nearest distance* that this node can get with regards to the designated location. The estimation is done based on the current location, and the moving direction of the corresponding node. A device responses upon a request for information with its predicted nearest distance towards the destination. Consequently, the devices with the closest distance are chosen as the next forwarding hops. Thereby, despite being designed as location-based forwarding, the MOVE framework does not consider a region, but rather a destination point. Furthermore, with the prediction for the nearest distance, the MOVE framework needs to assume a stable movement trajectory of the participating devices, which might not be suitable in the case of people-centric opportunistic network, in which the mobility of devices is uncontrollable.

Ma et al. [112] tackle the problem of forwarding towards a geographical region by identifying nodes with higher chances to visit the destination. The authors assume that the characteristic of the mobility pattern can be modeled as a Poisson process. Thus, by keeping track of the past visited locations, each device is able to construct a function to calculate the probability of this device to visit a designated location, which is expressed by the coordinates and a radius. To implement geographic based forwarding, Ma et al. duplicate several copies of a geocast message, which contain the coordinates of the geographical destination and broadcast the geocast messages in the network. Upon opportunistic contact, two devices can exchange the missing geocast messages, calculate the probability of the required location in the messages, and notify the source of the results. As a result, the source of a message can determine the candidate to carry the messages to the designated location based on the predicted probability value.

Geoopp [107] considers the geocast problem for opportunistic networks by dividing the area into different cells. For each cell, the authors calculate three probability values for each device, which indicate (i) the probability that the corresponding device will visit a particular cell, (ii) the probability that the corresponding device will stay long enough in the cell to deliver a message, and (iii) the probability that the corresponding device will be connected with other devices, which indicates whether this device will have sufficient resources and capacities for messages delivery. All three probability metrics are calculated by having each device track and record its past locations as well as connections with other devices within a cell. As a result, this consumes resources on the devices for storing information. Location-based geocast is realized, when two devices are connected to each other. Thereby, they will request to take over geocast messages intended for regions which belong to a cell that this node has a high visit frequency and a high probability of having sufficient resources for messages delivery.

A common point of the approaches mentioned above is that the location-based forwarding only utilizes a single copy of the geocast message. Using a single copy can save communication overhead. However, the location tracking and probability estimations on these devices suffer from computation overhead. Another direction to realize geocast is to utilize multiple copies of geocast messages. Soares et al. [178] propose a location-based forwarding concept named Geospray. Thereby, the authors use multiple copies of a geocast message to increase the delivery rate. Geospray combines geoopp [107] and the binary version of spray-and-wait [181]. The calculation of the visiting probability introduced in geoopp is used to estimate the time when the corresponding node will visit the destination. Geospray uses the binary spray-and-wait to replicate geocast messages. At each opportunistic contact, a device will give half of the replicated geocast messages to its neighbors until one copy of the message remains. In this way, several replicated geocast messages are disseminated in the network, increasing the chance to reach the destination. To restrict the overhead caused by replicating messages, geospray uses the *active receipts* to notify the devices in the network to remove copies of the delivered messages. When two devices meet each other, they exchange information about the geocast messages in their buffers. Thereby, they estimate the expected time to visit the destinations as being requested in the geocast messages. If a device is expected to visit a destination sooner that the current carrier, then the corresponding message will be given to the new carrier.

Cao et al. [27, 28] propose adaptive replication of messages for location-based forwarding with respect to a designated device. In the first step, when two devices meet each other, they will exchange information about their encounters with the destination. Based on the aggregated history encounters with the destinations from these devices, a future location for the designated device can be estimated. In [27], the authors propose to replicate the messages on devices which satisfy either one of two conditions (i) devices are approaching the destination faster, and (ii) devices will stay longer within a range near to the destination. To reduce delivery time and overhead, messages are prioritized in the buffer of each device according to a time-to-live value and to the distance of the carrier node towards the destination. A message will be prioritized more when the time-to-live value almost runs out and when the distance towards the destination is getting smaller. In this way, the messages are only replicated towards a predefined range. Extending this approach, in [28] the authors base their work on the spray-and-wait forwarding and only replicate a maximum number of messages into the geographical range towards the destination.

Similar to the approach above, Rajaei et al. propose GSAF in [156]. GSAF allows users to define a cast region flexibly by choosing multiple coordinates; consequently, defining a polygon form for a cast region is possible. With the defined geocast region for each message, the message is forwarded based on spray-and-wait. In the first step, a fixed number of copies of a geocast message are replicated and forwarded to the devices moving towards the cast region. As soon as a device reaches the cast region, it switches from replication using spray-and-wait approach to the epidemic flooding approach. Thereby, the messages are broadcast to all devices within the cast region, which ensures the availability of the messages within this region. When a device moves outside the cast region, it will automatically delete all corresponding geocast messages.

In this section, we reviewed state of the art for opportunistic forwarding in general and location-based forwarding in particular, since these approaches provide the communication basis for applications and services built on opportunistic networks. To wrap up this section, we provide some samples for applications and services built based on opportunistic forwarding to demonstrate the potential of opportunistic networks. In [194, 195], Tuncay et al. propose an approach for recruiting mobile devices to carry out sensing tasks. Thereby, the authors only consider homogeneous sensors in the network and use profile-cast [71] to disseminate sensing tasks to devices with visited locations closely matched with the intended sensing destination. Zhao et al. [221] design a solution to tackle the sensing area coverage problem. Thereby, epidemic flooding is utilized to exchange and synchronize the area which has been covered between two devices pairwise. Thus, a distributed coordination for crowd sensing is achieved.

In Fleanet [88], Lee at al. propose a marketplace for vehicular networks, which rely on epidemic flooding to facilitate dissemination of query and transaction between vehicles as sellers and buyers. Dealing with query dissemination for opportunistic networks using social metric, in [50], Fan et al. assume that the devices belong to at least one from many social communities and thus each device frequently visits its communities. This visiting pattern can reveal information about the future location of a device, thus can be leveraged for forwarding decision of a query.

Overall, opportunistic networks provide the communication medium to realize many types of services and applications. However, the communication in an opportunistic network is still host-based communication, therefore, special adaptation and handling on application layer to retrieve information are necessary. Recently, information-centric networking paradigm emerges, in which the communication is primarily based on a naming schema. Such a communication paradigm naturally facilitates information retrieval. In the next section, we discuss this networking paradigm in detail.

2.3 NAMED DATA NETWORKING — NDN

We base our mechanism to distribute crowd sensing tasks on the Named Data Networking paradigm due to its focus on addressing information. In this section, we first elaborate on the concept of information-centric networks in general and on Named Data Networking in particular.

2.3.1 *Background*

Even though location-based forwarding concepts of opportunistic networks can be adapted and utilized to support information retrieval, in essence, an opportunistic network is still based on host-to-host communication. Realizing the increasing importance of information retrieval in communication networks, recently, several research projects propose to shift the focus from host-to-host communication in existing networks to Information-centric Networking (ICN). ICN [3] has the following distinctive features: (i) *named data*, which is used to address the requested data/content/information instead of addressing the host, (ii) routing and forwarding based on naming conventions, which focus on retrieving the requested information, and (iii) in-network storage/caching to store named data, as well as request on each ICN capable node, which is similar to the *bundle* concept of DTN to decouple source and destination. Since ICN relies on self-encapsulated named data objects for communication, the requests for data are propagated to multiple data sources by default. Therefore, another advantage of the ICN concept is to leverage multiple data sources to balance the traffic of the network [14].

Named Data Networking (NDN) [217] is a networking architecture that implements the ICN concept, which is still being actively developed. NDN uses two types of packets for its communication to retrieve information, i.e., *Interest* packet and *Data* packet, which are shown in Figure 4. Interest packets are used to request information, while

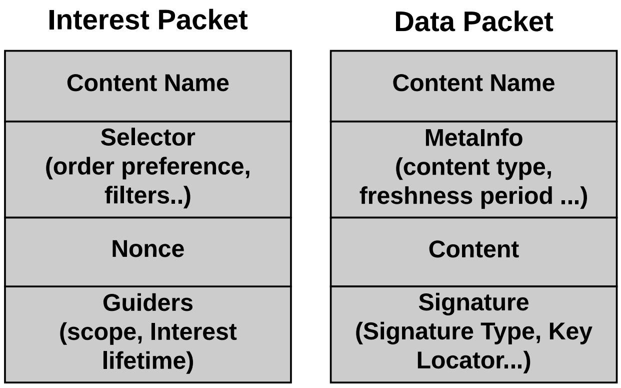


Figure 4: Illustration of interest and data packets used in NDN (adapted from [217]).

Data packets contain the information being requested. Interest and Data packets are identified through a naming convention, predefined by the NDN-based application. The process of requesting and retrieving information is illustrated in Figure 5.

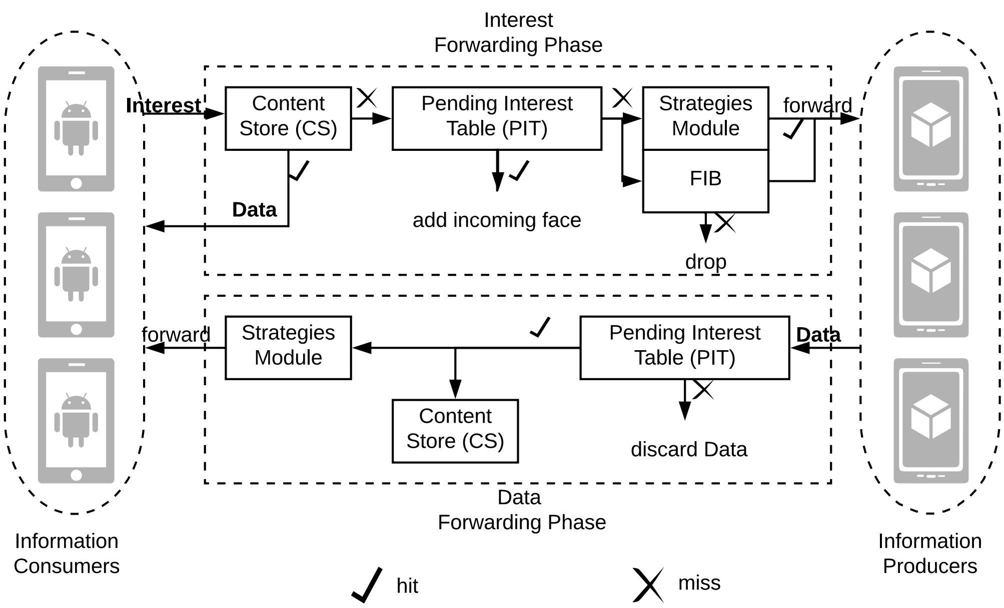


Figure 5: Illustration of interest and data packets forwarding in NDN (adapted from [217]).

Each NDN-capable device uses three data structures to realize the forwarding process, i.e., Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). Information retrieval in NDN is divided into two phases, i.e., *Interest forwarding phase* and *Data forwarding phase*. The whole process is triggered when an information consumer wants to retrieve a particular piece of information. Thereafter, the consumer sends out an interest packet with a predefined name. When an NDN node receives an interest packet, it first checks the CS, which serves as the storage to cache data packets. If a data packet exists in the CS with its name matched with the requested interest, this data packet will be forwarded back to the information consumer, along the way that the interest packet has traversed. In case a similar interest already exists in the PIT, the incoming interest packet will be dropped. If there is no matched data in the CS, the interest and the communication interface from which this interest is received will be added to the PIT. Thus, the PIT enables the data packets to backtrack towards the information consumers later. Afterward, the FIB and strategies module of the NDN node will be consulted for forwarding decision. When an interest packet reaches the *information producer*, which holds the requested data, the data packet will be constructed and forwarded back to the information consumers. This process is called *data forwarding phase*. By default, when data packets reach an NDN node, this node first checks to see if a matching interest exists in the PIT. Data packets will be dropped if no matched interest can be found to minimize possible redundant storage. If a matching interest exists, data packets will be stored in the CS. Thereby, if a similar interest comes in, the data can be served back directly to reduce the networking traffic to request for the same interest multiple times. NDN uses a hierarchical naming convention, e.g., */tu-darmstadt.kom/room218/temperature/_t18*, to identify interest and data. With such hierarchical naming convention, similar interest packets can be aggregated to save network traffic. Furthermore, an information consumer can request for information even when the naming convention is only known partially. To facilitate interest forwarding in networks with stable topology, devices can propagate the names of their possessed data over all network, so that each device can construct its FIB to guide the interest forwarding.

The performance of NDN in intermittent and disconnected networks has been studied and confirmed to be better than the legacy MANET in [190]. Tyson et al. [196] discuss in their work and point out several similarities between NDN and DTN/opportunistic networking, motivating for an integration of both networking concepts: (i) both support in-network storage, (ii) both decouple synchronous communication between sources and destinations, (iii) both rely on self-encapsulated network unit (message bundle, named data object). Despite these similarities, the challenge regarding the mobility of the information producer still has to be addressed to realize the integration of these two concepts [197]. In intermittently connected networks such as mobile opportunistic networks, the FIB table cannot always ensure a correct forwarding decision due to the rapid changes within networks. Furthermore, to realize NDN on mobile networks, one has to rely on broadcast using wireless transmission medium to forward interest and data packets [8]. As a result, broadcasting generates much overhead and collision of interest as well as data packets in NDN-based mobile networks. Consequently, forwarding strategies have to be designed carefully to allow successful delivery of data packets, considering potential packet collision. In the following, we review state of the art for interest packets forwarding in such dynamic mobile environments.

2.3.2 *Interest Forwarding in NDN-based Mobile Networks*

As aforementioned, interest forwarding to request for data on mobile networks is primarily based on wireless broadcast. In wireless ad hoc networking paradigms such as WSN, MANET, VANET, opportunistic networks, NDN-based forwarding approaches can be classified into two types according to [101]: (i) forwarding approaches that assume an existing route between information consumers and producers, and (ii) forwarding approaches that assume no available route. In general, for approaches that typically assume a stable network topology, the FIB can still be used to hold routing information based on requested name and to make the forwarding decision. For approaches with highly mobile devices, such as in a VANET and opportunistic networks, the FIB does not contribute to making forwarding decision, since rapid changes within the network can lead to bad forwarding decision. Therefore, further context information is required included in each broadcast packet to help devices make forwarding decision within the network. Regardless of the network stability, the challenge to avoid a *broadcast storm*, i.e., to minimize the interest collision as well as redundancy caused by the broadcast remains. The general idea to avoid a *broadcast storm* in NDN-based mobile networks is to introduce a defer timer, which lets each NDN-enabled device schedule packet broadcast by itself [8, 56, 158, 159].

The typical example of approaches considering stable topology can be found in the context of WSN research. Gao et al. [56] focus on enabling NDN-based WSNs. Since the topology of a WSN is rather stable, the FIB is utilized for forwarding decisions. The authors propose two modes to support interest forwarding, i.e., flooding mode and directive mode. The directive mode extends the interest packet with an ID, specifying the next forwarder. If this ID can be found in the FIB, the interest will be sent to the specified forwarder, accordingly. If the ID of the next forwarder cannot be found (due to the disappearance of nodes, or topology changes), the flooding mode will be used, which simply broadcast the interest further to all nodes. To avoid interest collision, a defer timer based on the residual energy of the corresponding node and the distance between the current node and the destination is calculated. Data packets will be forwarded back using the same interest forwarding route. Thereby, data packets contain hop count, and residual energy value of the information producers, which allow the devices along the path to learn and update news routing information in the FIB.

In [118], Meisel et al. propose an interest forwarding approach named listen first broadcast later (LFBL), which pioneers the integration of NDN concept in wireless ad hoc networks. The core idea of LFBL is to let each forwarder node to make its own decision, based on information embedded in each packet. Each packet contains source and destination IDs, which allow a node to determine the distance between these two. Upon receiving a packet, each node can add its distance to the destination into the packet before broadcasting (any distance metric can be used in the concept, e.g., hop count, geographical distance). The broadcast is triggered, only after each node overhears from other broadcasts, to confirm that it is an eligible forwarder with the smallest distance towards to destination. A random defer timer is used to schedule broadcast on each node, to avoid packets collision, however, how this defer timer can be determined, is not further discussed by LFBL.

Inspired by [118], other approaches for forwarding in NDN-based mobile networks follow the LFBL line of thought, thereby, specifying the calculation of the deter timer, distance metrics, as well as adding further context information in each broadcast packet. Amadeo et al. propose E-CHANET to realize NDN for MANET [9]. E-CHANET includes transmission rate information, the ID of consumer and producer, as well as, hop count as the distance metric in the packets. Similar to the approach of [56], E-CHANET relies on two modes, blind flooding if no forwarding information for a producer is available, and producer aware flooding, if the consumer can learn the route towards the producer from the past. In E-CHANET, nodes determine the defer time-based on a fixed defer time slot, multiplying with a random component. The random component is chosen to ensure the defer time of data packets is less than the defer time of interest packets. For every successful delivery of data packets, the consumers can learn about the hop counts, the transmission rate which causes congestion of the network. Thus, the transmission rate of interest packets and the estimated distance towards to producers can be adjusted by the consumers for future interest forwarding.

The aforementioned approaches provide a foundation to cope with the challenges when integrating NDN with wireless ad hoc networks but still focus on generic MANETs. In [38], Deng et al. distinguish between location-based interest forwarding, e.g., gas station for cars, and blind interest forwarding for VANET. To support location-based interest forwarding, the authors include the coordinates of the destination in interest packets. When a device receives a broadcast interest, it includes its current coordinates before rebroadcasting. Thereby, a device only rebroadcast if its coordinate is closer to the requested coordinate. The defer timer is calculated based on the distance of node and the destination, which ensures, that a node closer to the destination will have a lower defer time to speed up the transmission. To enable the *carry and forward* behavior similar to opportunistic forwarding, the authors rely on an explicit acknowledgement included in data packets. In case no acknowledgment is received after some time, the same interest packets will be rebroadcast again.

Navigo [58] is another approach for information retrieval for vehicular networks based on NDN. Thereby, Navigo focuses on retrieving information from multiple geographic-based sources. Navigo divides the map into several grid-based regions/-cells and introduces the abstraction GeoFace, which represent a single geographical region/cell. If the location of an interest is unknown, Navigo will flood the interest to all directions of the network. When data are found and forwarded back to the information consumers, the nodes along the way will learn about the association of the requested data with a GeoFace. The association between data and GeoFace is stored in the FIB table. Navigo relies on a Link Adaptation Layer to forward interest packets using Vehicle-to-vehicle (V2V) communication. Thereby, each device also chooses a random defer timer to delay broadcasting interest packets. For the forwarding strategy, with the association of data and GeoFace, each node can determine the cost to reach the destination, by applying Dijkstra's algorithm to determine the shortest path of the street segments, which leads to the destination from the corresponding node. Upon receiving an interest packet, a node only rebroadcast this interest, if the cost to reach the geo-destination is less than the cost of the previous hop.

Kuai et al. [82] consider a "delay tolerant interest forwarding" for vehicular networks. The destination for the interest forwarding in this work is the coordinate of an information producer. Thereby, the authors distinguish between the notion rebroadcast, which happens right after receiving an interest packet, and retransmission, which let the node store an interest after broadcasting, and forward it again later. In this work, the defer timer is determined through several priority values. A node closer to the destination of the producer will have higher priority. Between a forwarding device and the destination, the authors define an area characterized by an angle. Each forwarder node periodically broadcasts a special hello packet, which allows each node to determine its neighbors. Accordingly, each node can calculate a spatial priority value, based on the density of the neighbors, which lie inside the defined area among this node and the destination. A node will retransmit immediately if this spatial priority is greater than zero, which means that there are multiple nodes with closer distance to the destination.

Overall, the previously reviewed approaches represent the current state of the art for interest forwarding in NDN-based wireless ad hoc networks. In a nutshell, to enable location-based interest forwarding for this type of network, one needs to rely on broadcasting over wireless transmission medium. Therefore one has to consider the broadcast storm problem, the representation of geographical destination, distance metrics, as well as residual energy of participating devices. These metrics can be utilized for either the calculation of the defer timer or for the forwarding strategies to choose the next forwarding device. The reviewed interest forwarding approaches, however, do not consider the heterogeneity of information producers, which affects the quality of acquired information. In our work, we explicitly focus on the quality requirements of the information and on the heterogeneity of the participating devices.

2.4 DISTRIBUTED DATA PROCESSING

In this section, we review mechanisms to enable distributed data processing. Mainly, we focus on related research that allows for processing data directly within communication networks, i.e., distributed in-network processing, which is relevant for the second step of information retrieval. We refer to the definition of distributed data processing proposed by Enslow as follows:

Definition 2.2: Distributed Data Processing

"A distributed data processing [system] has five components:

- A multiplicity of general-purpose resource components, including both physical and logical resources, that can be assigned to specific tasks on a dynamic basis.
- A physical distribution of these physical and logical components of the system interacting through a communication network.
- A high-level operating system that unifies and integrates the control of the distributed components. Individual processors each have their own local operating system, and these may be unique.
- System transparency, permitting services to be requested by name only. The server does not have to be identified.
- Cooperative autonomy, characterizing the operation and interaction of both physical and logical resources." [47]

According to this definition, we discuss the related concepts with regards to the following aspects (i) the architecture and the communication, which correspond to the first two components of the definition, (ii) the abstraction of the function, which corresponds to the third and fourth components of the definition, and (iii) how the entities interact and coordinate the processing, which corresponds to the fifth and last component of the definition.

2.4.1 *Distributed In-Network Processing for Sensor Networks*

In Section 2.1, we have briefly elaborated on the WSN paradigm. In-network processing is an important concept introduced in WSN to execute operations, such as filter and aggregation on transmitted data directly within the network. This execution is carried out by the sensors located on the transmission path. The goal of in-network processing in WSNs is to minimize the communication traffic by reducing the data on the transmission path and consequently to preserve the energy of the whole WSN. One central question is which sensor nodes should execute which operations. This problem is termed *operator placement* [20]. In several cases, the topology of a WSN is stable, therefore, the placement problem can be solved by a centralized entity searching for an optimal solution, e.g, through Integer Linear Programming or greedy algorithms [30, 183]. However, in many cases and applications, the topology of WSN can also be changed. For instance, the WSN topology changes, if a sensor runs out of battery and disappears or if sensors are installed on moving objects such as animals. As a consequence, the operators placement in these cases needs to be solved in a distributed manner to allow for possible adaptation. In general, distributed algorithms for operator placement in WSNs rely on a periodic exchange or flooding of state information among local sensor nodes. A distributed placement algorithm starts with an initial placement, which will be refined and replaced with a new placement over time; a new placement is valid in case the total cost (e.g., total amount of data, energy consumption) for the old placement can be improved [20, 32]. To facilitate the deployment and the dynamic configuration for in-network processing or for operator placement, an abstraction schema is required. For instance, in [168], a rules-based abstraction is designed, which allows defining a processing goal, the conditions, and the rules to reach the processing goal through distributed processing by WSN nodes. Another example is T-Res [7], which defines a Representational State Transfer (REST) like Application Programming Interface (API) to enable reconfiguration of operations on the sensor nodes during runtime.

The concept of distributed in-network processing for sensor networks can be generalized. In a broader meaning, sensor networks can also be considered as networks with devices that are able to produce and process data. In this context, mobile devices in MANET and cars in VANET with the capability to sense/capture information from their surrounding also belong to this category. Hereby, Complex Event Processing (CEP) [165] emerges as an enabling technology for in-network processing of event streams. CEP is realized by installing an event processing engine on the participating devices and by deploying/placing functions/operators on these. Similar to in-network processing of WSN, algorithms for efficient *operators placement* to accomplish a processing goal and to achieve low latency while generating less overhead is desired. Thereby, centralized and decentralized placement algorithms for CEP in MANET have been studied [184]. Recently, Luthra et al. [109] also study a transition-enabled placement approach, which switches among different placement algorithms during runtime depending on the conditional changes, e.g., network load. Due to the application of CEP on mobile networks, the mobility of participating devices has to be considered. As a result, in addition to operator placement problem, *operator migration* is proposed as a solution to counter changes in the environment and to prevent degraded performance caused by these changes. Aiming to minimize the migration cost, Ottenwalder et al. [140, 141] propose to generate migration plans, which contain the policies to trigger migration and the migration targets. In this work, the migration targets are considered to be a network of infrastructure-based brokers. Mobility prediction for information consumers can be incorporated during the creation of a migration plan. Focusing on CEP operators migration for device-to-device communication in an Internet of Things (IoT) setup, Dwarakanath et al. [45] propose a lightweight *intermediate buffer* to store the processing states and the operations. This model can be used together with the mobility management to trigger a migration locally and directly among devices.

The concepts of in-network processing for sensor networks and of complex event processing as discussed in this chapter can enable distributed data processing. However, for opportunistic networks with rapid changes caused by mobility and without infrastructure based coordination such concepts cannot be directly used. In the next section, we discuss and review distributed processing mechanisms proposed for opportunistic networks.

2.4.2 *Distributed Processing in Opportunistic Networks*

In an opportunistic network, when two devices are connected to each other, they can not only exchange messages, but they can also provide their capabilities and resources as a service for the others [35]. Consequently, distributed processing can be realized upon each opportunistic contact by having a device send a task to offload the computation to the other device. Hereby, several aspects corresponding to Definition 2.2 have to be considered: (i) how can the participating devices interpret/understand the processing task, i.e., the need of an abstraction, (ii) how can the devices contribute to and coordinate the distributed processing under the the mobility of the devices and the rapid changes of the network, and (iii) how can devices make processing decision.

To provide abstraction and to facilitate the creation of distributed processing task, several researchers design and create middleware solutions for smartphones which enable computation offloading between two devices [36, 78]. The middleware solutions are often based on a client/server interaction model. A common abstraction among devices is realized through a stub/proxy residing on both the client and the server. A device that wants to provide a service, has to run the server instance, while a device that wants to use a service needs to run the client instance. As a result, a device in an opportunistic network might have to run both client and server instances, if it wants to offer and also utilize the resources in the network at the same time. Such model, however, restricts the flexibility of mobile opportunistic networks and constraints the services provisioning only between two one-hop neighboring devices. Recently, DTN-RPC [186] has been proposed based on Serval [57] with opportunistic forwarding capability. DTN-RPC utilizes the *message bundles* of Serval to realize a remote function call, which allows for asynchronous calls and thus decouples services provisioning between service consumers and service providers. Thereby, DTN-RPC supports two modes, i.e., direct offloading mode if the processing service provider is known and flooding mode to leverage mobility of devices to offload the processing task to a distant service provider. To allow for more flexibility when invoking remote processing, NFN [176, 193] introduces the *application-agnostic* named function to invoke the processing. NFN is designed as an extension for *information-centric networks*. NFN uses Lambda-expressions with a hierarchical naming convention of functions to allow the network to orchestrate and to coordinate the distributed processing. As an extension of *information-centric network*, the computation based on NFN can also be cached within the network, which benefits the execution time and reducing overhead. However, NFN does not support opportunistic networks with rapidly changing topology, no end-to-end path among devices, and intermittent connections between any two devices. Inspired by the concept of named functions from NFN, Graubner et al. [59] extend the Serval middleware [57] with hierarchical naming convention for function calls. Thereby, each message bundle of the Serval middleware, which contains the data and the named functions, can traverse through an opportunistic network. The execution of the function can be invoked upon finding a matching function on forwarding devices. Despite using the named function for data processing, the authors explicitly decouple the processing from networking. In our work, we integrate distributed processing into the data forwarding phase. Thereby, we leverage idle resources of participating devices during the result delivery phase of information retrieval.

In general, a device in an opportunistic network has to make a decision for the processing upon each opportunistic contact, i.e., whether a device can leverage resource of its neighbor to assign a processing task. This process is also called *opportunistic computation offloading*. Several strategies to support local decision making for opportunistic computation offloading have been studied. Two primary objectives of opportunistic offloading are to improve execution time and to reduce energy consumption of mobile devices. In [119], opportunistic task offloading on a cloudlet-enabled router is considered. Here, cloudlets are static devices with sufficient computing power in the proximity. The authors rely on probing several offloading tests to estimate the capacity of the cloud to make the offloading decision. Similarly, considering task offloading to cloudlet, Zhang et al. [219] rely on the workload and the accessibility to multiple cloudlets to make the offloading decision. In [214], Zeng et al. broadcast the processing tasks throughout the whole network with epidemic flooding. Devices that receive the complete data and possess enough computing resources will process. Besides having a device make a local decision, distributed processing can also be coordinated by several elected devices. In case the devices in an opportunistic network can form a cluster despite their mobility, the offloading decisions for mobile devices can be made by a local coordinator within each cluster [52, 64, 73]. Thereby, the participating devices have to notify the local coordinator of their current workload and capability profile, so that the coordinator can search for an optimal solution before assigning processing tasks. Such approaches, however, cannot sufficiently cope with the uncertainty caused by the mobility of devices in opportunistic networks.

Aforementioned opportunistic offloading approaches mostly focus on one-hop interaction and assume homogeneous processing tasks. To extract information from data, distributed processing might require to invoke multiple different functions and operations. Such requirement resembles the concept of *services composition*, in which services from different network entities are composed to create a new type of service. There have been several attempts to integrate and to enable services composition in mobile opportunistic networks. With respect to the mobility and the rapid changes of opportunistic networks, several objectives of opportunistic services composition have been considered, i.e., completion time, energy efficiency for devices individual devices, as well as load balancing for the whole network. The basic idea to enable services composition on opportunistic networks is to combine and unify the binding of a service provider in the composition and the execution of the services [60]. In [60], the authors, however, still rely on a request to search for a service provider, which cannot be guaranteed to be available until the binding and execution take place. Mascitti et al. [116] rely on a reactive approach. In this approach, instead of actively looking for a service provider, a device waits until the required service provider appears. To enhance the discovery of service providers, an overlay graph for service providers can be constructed dynamically to facilitate the discovery of service providers [29, 187]. To construct such overlay graph, upon each opportunistic contact, two devices exchange and extend their services graph. Therefore, these are able to look for the missing service provider. Based on the services graph, one can also select the composition path with the minimum network cost and a high probability of success. However, the handling of composition and services discovery as mentioned above still do not cope well the high mobility of devices in opportunistic networks. Without relying on an overlay services graph to look for a provider, several approaches delegate the complete composition and the responsibility of binding services to a service provider [33, 61, 167]. A service provider upon receiving the composition will execute the service accordingly. Thereafter, this provider searches for the next service provider (locally) and handover the whole composition hop-by-hop to the next service provider. In case, the next service provider cannot be found locally, the composition can be forwarded using epidemic flooding. Here, active announcement of services can be used to enhance the discovery of service providers. In this manner, services compositions and their corresponding requirements are realized for opportunistic networks.

The mechanisms discussed in this chapter considers individual aspects of distributed data processing, i.e., abstraction for functions, distributed coordination, services discovery. In our work, we propose a distributed processing model which consolidates all three aspects. Furthermore, we rely on autonomous decision of participating devices to cope with the rapid changes and uncertainties of opportunistic networks. Thereby, in our approach, the autonomous decision is not restricted only to the processing itself, but is also extended to controlling and management capabilities directly within network.

2.5 INFORMATION DISSEMINATION

The third step of information retrieval is *results delivery*. Since results delivery can also be considered as a subclass of information dissemination, we review relevant concepts of dissemination in this section. Thereby, we focus on dissemination on opportunistic networks.

Since forwarding and replicating messages are inherent for opportunistic networks, information dissemination is a natural application of an opportunistic network. In general, information dissemination resembles the concept and the goal of the publish/-subscribe model [19]. In a publish/subscribe model, several nodes act as publishers or information producers, that want to disseminate information to several interested nodes as subscribers. Hereby, for information dissemination on an opportunistic network, the information producers and consumers might not be aware of each other. In contrast, for *results delivery*, the disseminated information is the result/response of an explicit request for information retrieval. Therefore, in *results delivery*, the information producers might be aware of the identity of the information consumers. Accordingly, we can consider information dissemination from two perspectives: generic dissemination and consumer-aware.

To realize generic information dissemination on opportunistic networks, integrating publish/subscribe model seems to be a natural approach. To this end, Yoneki et al. [209] use the social metrics to realize publish/subscribe. Since mobile devices are carried by humans, which implies a potential social tie among devices, an opportunistic network can be divided into several communities. To detect a community within an opportunistic network, a distributed scheme such as k-clique [142] and the centrality metric are used. As a result, several communities can be detected and the central nodes of the community can be selected as the messages brokers according to the publish/subscribe model. The brokers exchange messages using normal opportunistic forwarding schemes to form an overlay network, which will be used for managing interest subscription and to disseminate information. The social metrics can also be used to determine the frequently visited locations of the participating devices, which indicate several rendezvous points to disseminate information [18]. Information dissemination also needs to adhere to the spatio-temporal requirements under the circumstances. For instance, in emergency response scenario, the information/notification are dependent on specific regions. Hence, information needs to be available within this region over a time period. To this end, Ott et al. propose the floating content [139], which relies on the replication of messages by devices entering the specified region to increase the availability of information. Based on the concept of floating content and considering emergency response scenarios, in [154], Psaras et al. propose to assign priorities to different types of information, e.g., authority notification, personal messages. The size of a floating region for information will be adjusted based on the priority. As a result, more critical information will have a greater floating region.

Dissemination of information on mobile opportunistic networks can also be realized with the information-centric networking approach in general and with Named Data Networking (NDN) architecture in particular. As an inherent feature, NDN-based mobile networks provide the in-network caching of data through Content Store (CS), which naturally supports dissemination by matching names of cached data packets with incoming interest packets. Due to the wireless broadcast used in NDN-based opportunistic networks, the data is available on multiple devices in the network; thus it increases the chance of successful dissemination. In [203], Wang et al. demonstrate a fast traffic information dissemination through the data forwarding phase of NDN-based vehicular networks. When a car detects an important event, e.g., traffic information, this car becomes an information producer that will disseminate this information through the data forwarding phase. The authors choose the defer timer for a data packet to avoid a broadcast storm and to allow the nodes with farther distance to broadcast faster, aiming to achieve faster and farther dissemination of traffic information in vehicular networks. Compared to [203], CODIE [4] is more consumer-driven. Thereby, the data forwarding is triggered only when receiving an interest packet, which explicitly requests for particular information. To reduce the caching overhead, the authors include a hop counter in each interest packet, which is used by the information producer to estimate an upper bound of hops count to forward the data back to the information consumer. In this manner, the authors achieve the goal that the data is disseminated to an area only near the information consumers. However, in this work, the authors use a simple street segment in which the cars are moving only in line with similar velocity. As such, the hop counts between the information consumer and the information producer do not change much over time. This assumption is in general not applicable in a highly mobile opportunistic network.

In case the data should be delivered to a specific information consumer, the opportunistic forwarding approaches introduced in Section 2.2 can be utilized. Thereby, the results of the information retrieval requests should be delivered in a timely manner; otherwise, the data might not be relevant anymore [126]. Most approaches, which aim to reduce delivery time, rely on the replication of messages to increase the chance of reaching the destination. However, for *results delivery* the data requested by an information consumer are only relevant for this one consumer. Hereby, replicating data introduces many redundant copies of the results in the network, wasting resources. In opportunistic networks, a consumer might also be mobile which makes it more challenging for *results delivery*. To cope with this problem, in [191, 192], Timpner et al. propose *"breadcrumb"* based routing to guide and forward the result of a query towards a mobile requester. If a mobile information consumer moves away from the original position, where it initially sends the query request, this information consumer will leave a "breadcrumb" message indicating the location where this information consumer leaves. A "breadcrumb" is realized based on floating content, which leverages devices in opportunistic networks to share and to maintain the information of the consumer's movement. Therefore, when the query request is completed, the result will be forwarded following the "breadcrumb" trace that the mobile information consumers leave behind. This approach, however, still generates overhead for keeping "breadcrumb" messages available to maintain the "breadcrumbs" trace. In our approach, we rely on mobility prediction to deliver results, which does not require an opportunistic network to maintain any tracking information.

Finally, as an extension for *results delivery*, the data can be pre-processed or aggregated during the forwarding process to reduce communication overhead. For instance, COUPON [222] considers the application of crowd sensing on opportunistic networks. Thereby, COUPON extends epidemic flooding and spray-and-wait forwarding with a fusion function. The goal of COUPON is to build a map of sensed data cooperatively using participating devices in the network, wheres each message in the network contains part of this map. As a result, it is possible to aggregate the contents of two similar sensing tasks into one thus reducing the number of replicated messages as well as reducing generated overhead. Despite the assumption of homogeneous data, COUPON demonstrates successful integration of distributed data processing and *results delivery* in opportunistic networks. In our work, we consider the heterogeneous capabilities of forwarding devices. Consequently, we are able to support more complex processing tasks during *results delivery*.

2.6 DISCUSSION AND IDENTIFIED RESEARCH GAP

In this chapter, we have provided background information of mobile opportunistic networks in Section 2.2. We reviewed related work with respect to the three steps of information retrieval, namely (i) location-based forwarding which can be utilized for sensing tasks distribution in Section 2.2.2, (ii) distributed data processing to preprocess or to analyze data directly within the network in Section 2.4, and (iii) information dissemination as well as *result delivery* in Section 2.5. Even though, most application scenarios such as emergency response situations rely on opportunistic networks as an alternative solution for infrastructure-supported communication, these applications still require certain quality guarantees to function properly. According to Steinmetz et al. [185], QoS requirements contain a set of parameters and conditions for an application/a service, that needs to be satisfied under any unforeseen circumstances which might occur during the service provisioning time. Providing QoS for applications/services require the applications/services to negotiate on the requirements with the relevant system components. The QoS negotiation consists of three steps: specifying the QoS requirements, determining the capacity of a system and the corresponding conditions, e.g., appropriate thresholds for QoS parameters, and *resource reservation* within the system to ensure the QoS provisioning. However, QoS negotiation is not feasible in mobile opportunistic networks, due to the following reasons [103]. (1) Due to the mobility of devices and the intermittent connectivity in opportunistic networks, there can be practically *no hard QoS guarantee*. (2) In general, there exists no end-to-end path among the sources and the destinations in opportunistic networks, which makes QoS negotiation impossible. Increasing delivery rate of messages in opportunistic networks alone cannot cover QoS requirements in general. (3) The QoS requirements for applications and services on opportunistic networks are not static and might change over time. E.g., the information might not be relevant anymore in the future, other operations might be added with higher priority than the currently running operations. Accordingly, we identify and raise the following questions for our work.

How to specify QoS requirements for information retrieval in opportunistic networks?

Most forwarding approaches, both generic forwarding as well as location-based forwarding proposed for opportunistic networks, mainly focus on optimizing successful message delivery to a single device or a group of destinations. However, these approaches are still limited to host-based communication and not designed for information retrieval. Accordingly, quality requirements of information retrieval cannot be translated directly to message delivery metrics. Related work on Named Data Networking, which is designed for name-based communication, show the potential to be integrated with opportunistic networks. In this research domain, most interest forwarding approaches, however, do not consider the heterogeneity and the dynamic mobility of information producers. Furthermore, general naming schemes in Named Data forwarding approaches in wireless networks still neglect the possibility to specify different granularity levels for quality requirements. Taking the changing QoS requirements caused by opportunistic networks into consideration, in our work we propose a naming scheme to specify multiple granularity levels for QoS requirements regarding information retrieval. We design an interest forwarding approach based on our naming scheme, which allows the forwarding devices to self-adapt towards satisfying the QoS requirements for information retrieval as fine-granular as possible.

How to enable distributed coordination and autonomous decision for processing? Related work for distributed processing in opportunistic networks such as WSN, Complex Event Processing, opportunistic offloading, and opportunistic services composition primarily focus on theoretical models to formulate an optimization problem and to solve this by a distributed algorithm. Several approaches that attempt to provide an abstraction which enables coordination and cooperation among participating devices, mostly rely on a middleware solution and a pairwise client/server communication model between two devices to offload computation tasks. Consequently, such abstractions restrict the flexibility of autonomous decision making by participating devices as well as restrict the integration and realization of distributed processing for mobile opportunistic networks. Following a message-oriented approach, which naturally fits for opportunistic networks, we propose an adaptive task-oriented message template to facilitate distributed coordination. Thereby, we explicitly support and promote autonomous decision making of participating devices; in the sense, that participating devices are allowed to change and to adapt the content of not only the data but also of the processing workflow, adding more flexibility for adaptation in opportunistic networks.

How to leverage heterogeneity and local strategies for supporting QoS requirements? The heterogeneity in resources and capabilities of participating devices can affect both forwarding and distributed processing. On the one hand, the heterogeneous resources regarding the residual energy, CPU, etc. have to be considered in order not to exhaust the available resources on participating devices. On the other hand, the heterogeneity in capabilities (sensors, available services) can be leveraged to improve the overall performance supporting QoS requirements. In all steps of information retrieval in an opportunistic network, a local strategy as well as an autonomous decision are required to handle the heterogeneity and to cope with the rapid changes of the opportunistic environment. Consequently, in this work we explicitly address the heterogeneity and the local decision for both forwarding as well as distributed processing. Regarding the forwarding phases (interest forwarding as crowd sensing task distribution and data forwarding as results delivery), we design our mechanisms to counter the heterogeneity of devices w.r.t. available resources in terms of the residual energy, mobility, and built-in sensors while satisfying quality requirements for information retrieval. With regards to distributed processing, we design several computation handover mechanisms supporting autonomous decisions and leveraging heterogeneity to satisfy quality requirements of a complex processing task.

In the upcoming chapters, we will present our forwarding and processing concepts, corresponding to three steps of information retrieval, i.e., crowd sensing tasks distribution, distributed in-network processing for data, and *results delivery*. All concepts are designed to utilize distributed coordination, autonomous decision, and local strategies to cope with the changing environment of opportunistic network, to uphold quality requirements.

3 SENSING TASKS DISTRIBUTION

Acquiring information through crowd sensing in general requires the participants to collect data from an Area of Interest (AoI) and to upload the collected data to a central cloud server. For example, PEIR [129] collects location data from people and combines with environmental data from other sources, such as weather data to assess their influences on the environment; Ushahidi [138] requires participants to report observed incidents during a crisis situation. In crowd sensing applications, the devices of the participants have to be triggered for data collection, with or without interaction of the participants. Regardless, the participating devices need to be aware of the conditions for the sensing campaign, such as predefined AoI, or the type of the requested data. To this end, the requirements for a crowd sensing campaign can be predefined in advance by the organizers. However, this does not fully utilize the potential of the various sensors available on participant's hand-held devices. A key factor to utilize the flexibility of crowd sensing is to create and distribute the so-called sensing task, which contains the requirements for the data that need to be collected by the participants. A sensing task can be created during run-time of the sensing campaign and distributed to the appropriate participant satisfying the conditions of the sensing tasks [62, 63]. In this manner, a crowd sensing campaign can be adapted to the changing needs by distributing the respective sensing tasks. Such feature is useful for an emergency response scenario, in which the information is critical for planning relief operations but constantly changing. In this context, one challenge arises: the communication infrastructure in an emergency situation, such as in disaster relief scenario, might not be available. Consequently, sensing tasks distribution in these situations has to be tailored towards the inherent specific requirements of such a scenario.

In this chapter, we tackle the problem of distributing sensing tasks to suitable participating devices in a decentralized fashion [131]. In Section 3.1, we discuss the general requirements of crowd sensing, as well as, specific requirements for a decentralized scenario, such as in emergency response situation. Based on the elaborated requirements, we present the core concept of our approach in general and, the corresponding system model in Section 3.2. Particularly, in this section, we elaborate on the communication architecture, the naming scheme, and the construction of the so-called *Interest packet* required for distributing the sensing tasks in decentralized manner. Based on the components of the system model, we provide details of the designed forwarding mechanism as the enabler for the sensing tasks distribution in Section 3.3. The proposed forwarding mechanism is designed to counter mobility and the constantly changing conditions. The concept presented in this chapter focuses on the distribution of the sensing task; as such, the evaluation presented in Section 3.4 is constructed to allow us to analyze the sensing tasks distribution in depth. Together with the distributed processing concept in Chapter 4 and the concept for results delivery in Chapter 5, information retrieval in an opportunistic network can be achieved. Thereupon, we will present a consolidated evaluation of sensing tasks distribution and results delivery later in Chapter 5.

3.1 REQUIREMENTS AND CHALLENGES

Due to the fact that crowd sensing relies on mobile devices carried by humans to collect data, the quality of crowd sensing relies largely on human factors, e.g., how the participants move, which capabilities their devices provide. Since most of the human factors are uncontrollable, the requirements needed to ensure high quality of a crowd sensing campaign have to be considered. The following requirements for a general crowd sensing paradigm can be derived [62, 111, 215]:

Quality of Information (QoI): The objective of crowd sensing is to collect data from an AoI to extract valuable information. Quality of information is therefore an inherent requirement. Distributing the sensing tasks to appropriate participants contribute to improving the overall quality of crowd sensing. Thereby, the quality of the crowd sensing tasks can be characterized by "4W1H—what, when, where, who, and how" [215]. *What* refers to the type of data that need to be collected; this again depends primarily on the type of sensors available on the mobile devices of the participants. *When* indicates the temporal requirement, at which time the data need to be captured. *Where* refers to the spatial requirement of the AoI. *Who* indicates the requirements of the participants, which need to be considered before assigning sensing tasks. In this regard, mobility of the participants plays an important role, since the mobility is uncontrollable and can affect both the spatio-temporal requirements. *How* refers to the specific execution of the sensing tasks on the participating mobile devices, e.g., frequency of the data sample rate.

Cost of crowd sensing campaign: Since crowd sensing leverages mobile devices of participants, the deployment cost of a static/special-purpose sensor hardware can be saved. However, the cost for executing the crowd sensing application (e.g., network cost to allocate the sensing tasks, energy cost on participating devices when performing data sampling) in general often conflicts with the quality of information [202]. Often, to acquire enough measurements to cover an AoI while satisfying the QoI requirements, a large number of participants might be required, which in turn requires the distribution of large number of sensing tasks, indicating more cost. As a result, one of the main challenges is how to allocate the sensing tasks to a minimum number of participants to ensure QoI.

Incentive for participants: One common aspect that can be observed from the discussion of the aforementioned requirements is the number of participants. In order to achieve high participation of humans (their mobile devices) for a crowd sensing application, there needs to be an incentive. Three categories are identified: *entertainment, money,* and *services* [218]. *Entertainment* refers to crowd sensing in form of a multiplayer game, which motivates the players to provide crowd sensed data. *Money* refers to the monetary payment, which the participants get in return for providing data. The last category, *services*, refers to the form of providing data for a group of users, or community, such that all members of this group, or community can benefit from the crowd collected data. The last category aligns well with the considered emergency response scenario in our work, in which the crowd collected data can be used to provide emergency services such as *Person Finder*[1], or to generate a heat map of affected regions, which enhances the relief operations. We assume, that the participating devices want to contribute their resources according to this incentive category.

In addition to the requirements discussed above, the crowd sensing application in emergency response situations introduces the following challenges, making it more challenging to satisfy the aforementioned requirements.

Decentralized communication infrastructure: The communication infrastructure plays a vital role in both distributing sensing tasks, as well as in collecting the data [62]. However, in an emergency situation, such as in disaster relief scenario, the communication infrastructure tends to be a *hybrid* communication infrastructure. As such, not all devices have access to the Internet; however, mobile devices can still communicate directly through (WiFi) Ad Hoc communication. Overall, the network of the participating devices in the crowd sensing application in this case might be highly partitioned. Furthermore, the mobility of participants also renders the ad hoc network highly dynamic and unstable, adding to the challenge of assigning the sensing tasks.

Heterogeneity of participating devices: Users participating in a crowd sensing application possess different mobile devices, hinting at heterogeneity in capabilities [89]. Accordingly, not all mobile devices can fulfill the requirements for the QoI of the application. Given the decentralization caused by partitioned network in an emergency response situation, an overview of participating devices is not always available; it is therefore even more challenging to find, and to assign the sensing task to the right participants. Besides heterogeneous capabilities, the participating devices might also possess heterogeneous resources, such as their current energy level. This type of heterogeneity has to be considered when assigning sensing tasks, since the quality of crowd sensing also depends heavily on this factor, e.g., assigning sensing tasks to a device with low energy level can potentially lead to low data availability, for such device can fail to capture data when running out of battery.

3.2 SYSTEM MODEL

With the requirements discussed in the previous section, we now detail the system model for crowd sensing. We focus on designing a solution for crowd sensing tasks distribution in highly dynamic mobile opportunistic networks, considering the disaster relief scenario. Based on the common taxonomy of a crowd sensing architecture shown in [63], our system model follows a hybrid communication architecture, which is more suitable for disaster relief scenario [93]. The system model consists of a *tasking server*, which in an emergency response scenario serves as an interface for the authority to define and organize the overall crowd sensing campaign; of several *gateway devices* with access to the *tasking server* despite the possible impaired communication infras-

1 https://google.org/personfinder

tructure; and of mobile devices with built-in sensors, which can provide requested data for the sensing tasks, and which can also be used to exchange information locally through opportunistic ad hoc communication. With respect to the system model and the considered scenario, we make the following assumptions:

- Without loss of generalization, we assume that the sensing tasks, intended to request data from an AoI, cannot be distributed directly to mobile devices inside the AoI. This assumption is due to the typical decentralized communication of the disaster relief scenario caused by impaired communication infrastructure.
- The heterogeneity of participating devices implies, that not all devices possess the same set of built-in sensors. This also means that not all devices are able to capture the data type as requested. Therefore, we assume that only a subset of devices are capable of satisfying the sensing task's requirements.
- As discussed through the requirements for crowd sensing, crowd collected data in a disaster relief scenario can be used to provide emergency services, which are beneficial for all concerned people. Thus, we assume that the participating devices collaborate with each other without malicious intention.
- As common in modern mobile devices, we assume that the participating devices can determine their current locations accurately, either through built-in GPS, or through collaborative local monitoring mechanism such as in [162].

Our goal of distributing sensing tasks thus can be divided into two sub-objectives: (i) the first sub-objective is to successfully carry the sensing tasks to the requested AoI, considering the challenging communication architecture in emergency response situations; (ii) the second sub-objective is to search for the mobile devices within the requested AoI, that possess the right capabilities to capture the requested data, taking the heterogeneity into consideration.

3.2.1 *Hybrid NDN based Communication Architecture*

To achieve the two sub-objectives, we propose a hybrid communication architecture for our crowd sensing solution, which we base on the NDN paradigm. The decision is made due to its focus on addressing *information*, instead of addressing hosts, and its inherent characteristic to support highly dynamic networks. The suitability of the NDN paradigm to collect information through crowd sensing is also confirmed by Moreira et al. [127]. Additionally, Bouk et al. [21] point out that the NDN paradigm can co-exist with the existing networking technology, such as IP-based networks. Hence, for our communication architecture, a hybrid solution of NDN and widely deployed IP-based networks is proposed. Our hybrid architecture is illustrated in Figure 6.

To achieve the sub-objective of bringing the sensing tasks nearer to the AoI, the *tasking server* relies on several gateways to first inject the sensing task into the opportunistic ad hoc network, formed by participating mobile device. We consider the tasking server as an abstract interface for the crowd sensing organizers, therefore we

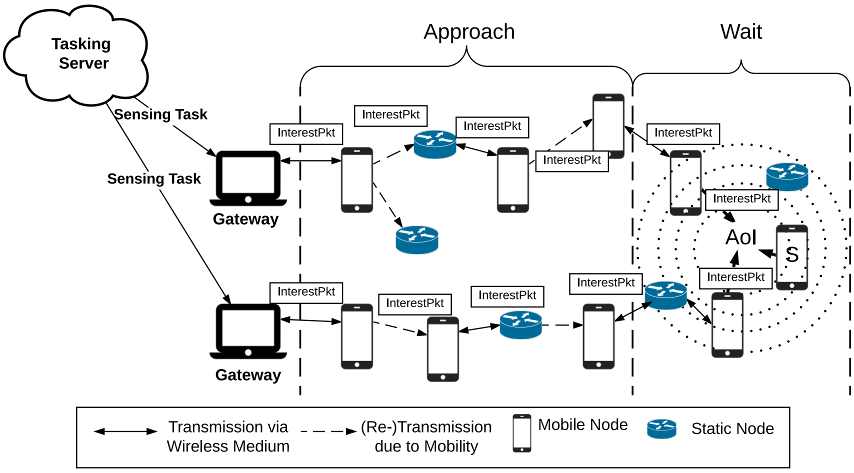


Figure 6: Illustration of our hybrid NDN based communication architecture for distributing sensing tasks (image from our publication in [131])

assume the availability of this entity in the system. As the name suggests, gateways are devices which still have connection to the tasking server, despite the critical network condition in emergency situations. Such gateways can take on several forms. For instance, in [132], an on-board component for firefighter's truck is built based on a Raspberry Pi device, which utilizes LTE to maintain a connection between each truck and the tasking server. Meurisch et al. [120, 121] analyse the coverage of home routers within a city, and show that the home router can be upgraded to a cloudlet, which can provide communication in emergency situation; there is a high possibility, that several devices from the high number of router-upgraded cloudlets are still able to uphold connection to the backbone network, thus is able to communicate with the server. Last and not least, the mobile devices themselves might still be able to communicate with the tasking server in emergency situation, thus a gateway device can be chosen from such devices through gateway selection mechanisms, e.g, using clustering mechanisms [161, 164]. All in all, it is possible to first inject the sensing tasks from the server through several gateway devices nearest to the AoI into the emergency opportunistic network.

Beside the role of a client to receive the sensing tasks from the tasking server, a gateway device also assumes the role of an *information consumer* according to the NDN networking paradigm. As an *information consumer*, a gateway device is in charge of constructing an *interest packet*, which contains the requirements requested from the sensing tasks. Next the gateway device—*information consumer* propagates the *interest packets* within the NDN-based opportunistic ad hoc network. The goal of the interest propagation is to successfully transport the interest to the mobile devices capable of providing requested data within the AoI. This process represents the distribution of sensing tasks in an opportunistic network. Accordingly, such mobile devices assume the role of *information producer* for the NDN-based network. After receiving the data back from the *information producer* as requested, the gateway devices can forward the data back to the tasking server for overall aggregation.

Under the assumption, that the gateway devices are not located directly inside the AoI, and that the participating devices are highly mobile, the mobility of the participating devices can be leveraged to bring forward the Interest packets from the *information consumer* to the AoI. Hereby, the participating devices (*consumers, forwarders, and information producers*) can communicate with each other through ad-hoc WiFi connections. Corresponding to the two sub-objectives of our design, we propose to construct two phases for the Interest forwarding. These two phases are illustrated in Figure 6 as *approach phase* and *wait phase*. The goal of the *approach phase* is to first leverage the mobility of the forwarding devices for bringing the Interest packets as near to the AoI as possible; while the goal of the *wait phase* is to counter the mobility of *mobile information producers*. To this end, our approach is inspired by the floating content concept introduced in [139]. In the *wait phase*, interest packets are *floated*, i.e., bound geographically in close proximity of the AoI, aiming to increase the chance of reaching the appropriate *mobile information producers*, which can provide the requested data as soon as possible. Binding sensing tasks to geographical region can also increase the chance that, when an information producer leaves the region the sensing tasks can be offloaded to other producer. Consequently it can improve data availability and avoid missing potentially critical information. Similarly, the idea of creating and maintaining sensing tasks geographically can be found by the work of Campbell et al. in [106]. However, their work relies on communication infrastructure, while our approach is designed for and focuses on decentralized mobile opportunistic networks.

Having elaborated on the system model in general and on the hybrid communication architecture in particular, in the following we will describe the *naming scheme* of the interest packets, which is used to express the requirements of the sensing tasks and the construction of the interest packet, which contributes to facilitate the interest forwarding. Using the proposed naming scheme and the interest packets construction, we then detail our concept of two-phase interest forwarding.

3.2.2 *Naming Scheme*

In NDN-based networks, each interest packet contains the request for data, which is identified through a predefined naming scheme. The naming scheme serves for two purposes: (i) identification of the requested data, and (ii) guiding information for forwarding decision as defined in FIB. However, using a naming scheme and the FIB alone is not sufficient to make forwarding decision in an opportunistic network, since, the entries in the FIB are not reliable when an end-to-end path is not always possible. Hence, further information is needed in interest packets to guide the forwarding process, which will be discussed in the following section. In this section, we focus on the naming scheme for identification of requested data.

For crowd sensing applications, the data requested has to satisfy several QoI requirements. As discussed in Section 3.1, the requirements for a crowd sensing task are characterized by five main quality dimensions [215], i.e., "*what to measure, where to measure, when to measure, who to measure, and how to measure*". Accordingly, the naming scheme for an interest packet needs to cover these dimensions as well. We propose the naming scheme for an interest packet of crowd sensing applications as follows:

Definition 3.1: Naming Scheme for Interest Packet

/CrowdSensing/<geographical-information>/<sensor-type>/<time>

In the above definition, *<Geographical information>* represents the requirements of the requested AoI. Hence, the information on *where to measure* is given. *<Sensor type>* represents the requested data type; i.e., the *information producer* should possess the corresponding sensor, in order to trigger the data collection. Thereby, the information on *what to measure or who should measure* is given. *<Time>* incorporates time related requirements, i.e. specific time *when to measure*. Further instruction for the execution of sensing tasks, such as the frequency of the measurement are given in *how to measure*, which is part of time related requirements. Overall, all quality dimensions for a crowd sensing application can be captured using the proposed naming scheme.

With regards to the geographical information, two options are possible to represent the location of an AoI, i.e., the AoI can be represented through named association, such as street address, or through coordinates of the location. The first option is not always possible in a decentralized environment, in which the participating devices despite being able to determine their own location, will be unable to interpret an address association. Consequently, for the considered scenario, the geographical information of the AoI should be represented through its coordinates. Furthermore, since in NDN networks, each participating device maintains a CS table to cache the data, the naming scheme can also be used to match the data temporarily stored within the forwarding devices, reducing the data delivery time. For this reason, we use the geographical representation proposed by Pesavento et al. [148]. Given (x, y) as coordinates of the AoI, the authors use a Cantor pairing function to transform the pair (x, y) into an ordered sequence of numbers $c_1..c_n$. The Cantor pairing function takes two numbers as input, and is calculated as follows:

$$f_C(n_1, n_2) = \frac{1}{2}(n_1 + n_2)(n_1 + n_2 + 1) + n_2 \tag{1}$$

The ordered sequence of numbers $c_1..c_n$ is determined by applying the Cantor pairing function on all aligned digits of the coordinate x, y. In case, x, y have different digits length, zeros are padded until both of them have the same length. As such, the *<Geographical information>* assumes the form $/c_1/../c_n/$. Since the name of the data in NDN is matched based on the longest prefix, such representation allows a crowd sensing application to define how accurate the data should be matched with respect to the location of the AoI.

Furthermore, longest prefix matching in NDN also allows us to define sensing request for different needs. Several examples to demonstrate the flexibility of the proposed naming scheme to express different quality requirements for crowd sensing requested data are given in Table 1.

With the naming scheme, we accomplish the objective of defining the sensing tasks and their quality requirements. In the following section, we further elaborate on

Table 1: Examples of named interest packet for crowd sensing application

Named Interest	Meaning
/CrowdSensing/	request for all information
/CrowdSensing/c1/../cn/	request for all information covered the AoI with coordinates corresponding to the sequence $c_1..c_n$
`/CrowdSensing/c1/../cn/ SensorX`	request for data type X (collectible through sensor X), from the requested AoI
`/CrowdSensing/c1/../cn/ SensorX/date/time/f`	request for data type X, from the requested AoI, for particular data, time, with sampling frequency f.

the construction of the whole interest packets, aiming to aid the interest forwarding decision in opportunistic ad hoc networks.

3.2.3 *Interest Packet Construction*

Forwarding interest packets in our scenario has to rely on the *store, carry, and forward* paradigm of opportunistic networks, in which each device after receiving an interest packets, can store it locally and later forward the interest packets to other devices with better chance to reach the intended destination. This is due to the fact, that an end-to-end path between the *information consumers* and the *information producers* is not always available; the topology of the network keeps changing over time. Therefore, the common approach of NDN to propagate an interest through network (inter-)faces which is predefined in the FIB table of each device, is not possible. Instead, each interest packet will be (re-)broadcast through the same network interface of the devices. In case of mobile opportunistic ad hoc network, interest packets will be (re-)broadcast through WiFi network interface. As a result of such uncoordinated interests (re-)broadcast, much overhead, redundancy and collisions are generated, potentially leading to an overall degradation of network performance [210]. To cope with this problem, distributed coordination for interest packet forwarding is desired. As a consequence, we rely on each device to share its context information, such that this context information can be utilized by other devices to improve the forwarding decision. For this purpose, we leverage the attribute fields of the interest packet to embed context information of each device before broadcasting the interest packets. The modified interest packet with the additional attribute fields is illustrated in Figure 7.

The context attributes embedded into interest packets are: (i) the current distance of the corresponding devices, before broadcasting the interest packets, (ii) the maximum distance from the *information consumers* to the AoI, as observed by the corresponding devices, and (iii) the total number of interest packets, that have been broadcast by the corresponding devices up until now. When a device receives an interest packet, it can extract these three attributes from the packet. Combining the extracted attributes with the current local context information (such as residual energy level, characteristics of the current movement), a device can make a forwarding decision in favor of reducing

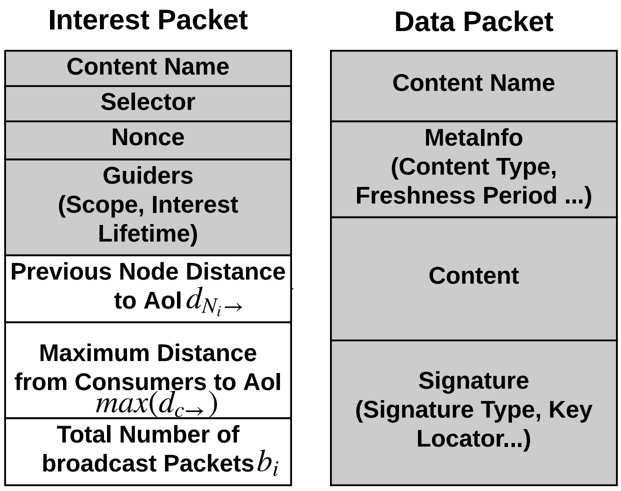


Figure 7: Data packet and modified interest packet with 3 additional attribute fields (image from our publication in [131]). Default fields are adapted from [217].

interest over-broadcasting, reducing the time-to-find *information producer*, and conversing the energy of the whole opportunistic network. We discuss the detailed use of the context information for forwarding decision in the next section.

3.3 TWO-PHASE INTEREST FORWARDING

We design our interest forwarding solution as a concept to distribute sensing tasks to the appropriate participating devices, to acquire data from a predefined AoI. In this section, we first elaborate on the overall concept and the corresponding workflow. Thereafter, we provide details on our context-aware forwarding, aiming to satisfy quality requirements of the crowd sensing application, while generating minimum overhead.

3.3.1 *Overall Concept and Workflow*

As previously discussed in Section 3.2, we first utilize gateways to inject the sensing tasks to the NDN-based opportunistic ad-hoc network. Based on the quality requirements predefined by the tasking server, the gateways in the role of *information consumers*, constructs the corresponding interest packet. Thereafter, the *information consumers* start broadcasting the interests into the network. When a forwarding mobile device receives the interest packet broadcast from its neighbors, it acts accordingly to the NDN based paradigm, and checks to see if the requested data is available in the CS table. In case the longest prefix of the name of the requested interest and the name of the stored data can be matched which indicates the data in the CS satisfy the quality requirements of the *information consumers* (cf. sample in Table 1), then the data can be forwarded back to the *information consumers* directly from the corresponding devices. In case no matched data can be found stored in the forwarding device, this device will further broadcast to forward the requested interest towards the AoI. Since broadcasting interest in an uncoordinated manner will generate much overhead, leading to collision and high energy consumption at each participating devices, there needs to be a distributed coordination among the devices (a centralized coordination is not always possible in the considered scenario, and is also inefficient for highly mobile networks). To achieve distributed coordination in the interest forwarding process, we rely on decentralized self-organizing patterns [37]. Decentralized self-organizing patterns are proposed and designed to coordinate autonomous software agents in a distributed manner. The basic idea is to utilize virtual force fields to guide the behavior of the devices. Within the decentralized self-organizing patterns, we utilize the *SPREAD* and *ATTRACT* gradient patterns to guide the interest forwarding decision. These are illustrated in Figure 8.

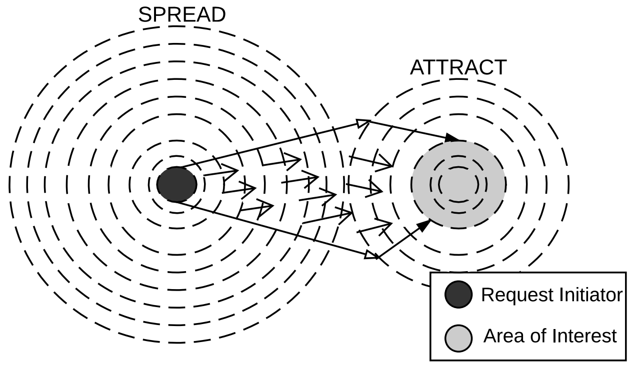


Figure 8: Combining *spread* and *attract* gradient patterns to guide interest forwarding decision. (image from our publication in [131])

The illustration in Figure 8 also corresponds to the two phases discussed with the hybrid communication architecture in Section 3.2.1. The *approach phase* in the interest forwarding first aims at forwarding the interest packets closer to the AoI; the *wait phase* focuses on dealing with the mobility of the *information producers*. As soon as the interest packets approximate the location of the AoI, the *wait phase* is triggered to bind the interest packet close to the AoI, aiming to *wait* for *mobile information producers* to appear in the AoI, that can provide the requested data. Accordingly, the *spread* gradient field is applied around the *request initiator*, a.k.a., *information consumer*; while the *attract* gradient field is applied around the AoI. The combination of these two patterns results in a directive forwarding flow, performing the desired behavior to (i) first push the interest as far away from the request initiator as possible to look for the *information producer*, while (ii) regulate the direction of the forwarding by pulling the interest packet towards the AoI, as these get closer to the AoI location. Altogether, our combining pattern can solve the two sub-objectives introduced in Section 3.2, leading to a decentralized concept to distribute sensing tasks in opportunistic networks.

The details of our two-phase forwarding mechanism will be given next.

Table 2: Parameters for interest forwarding concept and their meaning

Parameters	Meaning
$Time_{DeferSlot}$	maximum defer time
DT_{Int}	defer time for interest packet
DT_{data}	defer time for data packet
R_{BZ}	radius of the *buffer zone*
R_{AoI}	radius of the AoI
s_i	current speed of device N_i
e_i	current energy level of device N_i
b_i	current number of broadcast interest packets by device N_i
b_p	number of broadcast interest packets from neighboring devices, as extracted attribute from a received packet
$d_{N_i\rightarrow}$	the current distance between device N_i and AoI
$d_{N_p\rightarrow}$	the current distance between device neighboring devices and AoI, as extracted attribute from a received packet
$d_{c\rightarrow}$	the distance between consumer and AoI
$\overrightarrow{md}$	vector representing moving direction of device N_i
$\overrightarrow{N_i}$	vector representing straight direction from device N_i towards AoI

3.3.2 *Context-Aware Two-Phase Forwarding*

In a nutshell, the two-phase interest forwarding concept relies on an *approach phase* to first forward the interest towards the direction of the AoI and second to "float" the interest packets geographically around the AoI. To realize the two-phase interest forwarding concept, based on the decentralized self-organizing pattern as discussed in the previous section, we introduce the notion of a *buffer zone*. The goal of the *buffer zone* is to serve as the area to float around the AoI to wait/buffer for the appearance of the appropriate *information producers*. In Figure 8, the *buffer zone* corresponds to the *attract* gradient field. Respectively, outside the *buffer zone*, devices broadcast interest packets according to *approach phase*; as soon as devices enter the buffer zone, the interest will be broadcast according to *wait phase*. All the parameters required for the two-phase forwarding mechanism are summarized in Table 2. The pseudo algorithms to process interest packets according to *approach phase* and *wait phase* can be found in algorithm 1 and 2 respectively.

Algorithm 1 : Interest packet processing of device N_i during the *approach phase* with R_{BZ} as the radius of the *buffer zone*

Input : Interest packet containing context attributes
Result : Forwarding decision; Interest drop or rebroadcast with defer time

begin
    $(d_{N_p\rightarrow}, d_{c\rightarrow}, b_p) \longleftarrow \mathtt{extractContextAttributes(InterestPkt)}$;
    $d_{N_i\rightarrow} \longleftarrow d(N_i, AoI)$;
    $\overrightarrow{md} \longleftarrow N_i'$s current moving direction;
    $s_i \longleftarrow N_i'$s current speed;
    $e_i \longleftarrow N_i'$s current energy;
    $b_i \longleftarrow N_i'$s total broadcast packets;
    $d_{max} \longleftarrow max(d_{c\rightarrow})$;
    if *Data with matched name found in Content Store* **then**
        $DT_{data} \longleftarrow Time_{DeferSlot} * (\frac{d_{N_i\rightarrow}}{d_{max}}) + T_{Random}$;
        Schedule to broadcast Data after DT_{data};
    else
        if $(d_{N_i\rightarrow} < d_{N_p\rightarrow})$ *and* $(\angle(\overrightarrow{md}, \overrightarrow{N_i}) < \angle_{threshold})$ *and* $(e_i > e_{threshold})$ *and* $(s_i = 0$ *or* $s_i > s_{threshold})$ **then**
            $\mathtt{isForwarder} \longleftarrow \mathtt{TRUE}$;
        if $\neg$ *isForwarder* **then**
            Drop Interest;
        if $d_{N_i\rightarrow} > R_{BZ}$ **then**
            if *Interest* $\longleftarrow$ *Find(PIT)* **then**
                Discard incoming Interest;
                Increase Interest's lifetime and update PIT;
            else
                Add $(d_{N_i\rightarrow}, d_{max}, b_i)$ to Interest packet;
                Insert Interest to PIT;
                $DT_{Int} \longleftarrow Time_{DeferSlot} * (T_d + T_e + T_s + T_{md}) + T_{Random}$;
                if $b_i < median(b_p)$ **then**
                    $DT_{Int} \longleftarrow DT_{Int} * \frac{b_i}{median(b_p)}$;
                Schedule to broadcast Interest after DT_{Int};
        else
            proceed to *Wait phase*;

Algorithm 2 : Interest packet processing of device N_i during the *wait phase* with R_{BZ} as the radius of the *buffer zone*

Input : Interest packet containing context attributes
Result : Forwarding decision; Interest drop, rebroadcast, and replication with defer time

begin
  $(d_{N_p \rightarrow}, d_{max}, b_p) \longleftarrow \mathtt{readContext(InterestPkt)};$
  $d_{N_i \rightarrow} \longleftarrow d(N_i, AoI);$
  $b_i \longleftarrow N_i^{\prime}s\ \mathtt{total\ broadcast\ packets};$
  if $d_{N_i \rightarrow} < R_{BZ}$ **then**
    if *Data with matched name found in Content Store* **then**
      $DT_{data} \longleftarrow Time_{DeferSlot} * (\frac{d_{N_i \rightarrow}}{d_{max}}) + T_{Random};$
      Schedule to broadcast Data after DT_{data};
    else
      if *Interest* $\longleftarrow$ *Find(PIT)* **then**
        Discard Interest;
        Increase Interest lifetime and update PIT;
      else
        if $d_{N_i \rightarrow} < R_{AoI}$ **then**
          $d_{max} \longleftarrow R_{AoI};$
        else
          $d_{max} \longleftarrow R_{BZ};$
        Add (d_{max}, b_i) to Interest;
        Insert Interest to PIT;
        $DT_{Int} \longleftarrow Time_{DeferSlot} * (T_d + T_e + T_s + T_{md}) + T_{Random};$
        if $b_i < \mathrm{median}(b_p)$ **then**
          $DT_{Int} \longleftarrow DT_{Int} * \frac{b_i}{\mathrm{median}(b_p)};$
        $n_{REP} \longleftarrow n_{max} * \frac{R_{BZ} - d_{N_i \rightarrow}}{R_{BZ}};$
        for $i \leftarrow 1$ *to* n_{REP} **do**
          Schedule to broadcast Interest after DT_{Int};

In order for a participating device N_i to determine, when to switch to *wait phase*, this device can compare its current distance towards the center of the AoI ($d_{N_i \to}$) and the radius R_{BZ} of the *buffer zone*. Under the assumption, that each device can determine its current location in longitude, latitude coordinates ($long_{N_i}, lat_{N_i}$) via GPS, these values can be transformed into the Cartesian coordinates (loc_{x_i}, loc_{y_i}) by the Mercator formula [113]. Since the Cartesian coordinates ($loc_{x_{AoI}}, loc_{y_{AoI}}$) of the AoI are also given in the interest packet, we use the Euclidian distance for the current distance from device N_i towards AoI calculated as:

$$d_{N_i \to} = \sqrt{(loc_{x_i} - loc_{x_{AoI}})^2 + (loc_{y_i} - loc_{y_{AoI}})^2} \qquad (2)$$

If $d_{N_i \to} > R_{BZ}$, then forwarding behavior according to *approach phase* is triggered. The pseudocode for the *approach phase* is provided in algorithm 1. The radius size of the buffer zone R_{BZ} is a configurable parameter for the forwarding mechanism, which can be set by the tasking server or the gateway devices. However, it is impossible to specify an optimal value for R_{BZ} in advance, since in a mobile opportunistic network the size and the topology of the network can change rapidly. We will show later in the evaluation presented in Section 3.4, that the buffer zone radius affects the performance of the sensing tasks distribution with regards to the density of the network; thereby, we present the trade-off between buffer zone radius and other performance metrics.

As discussed in Section 3.2, interest packets in an NDN-based opportunistic ad hoc network have to be (re)-broadcast, which relies on the *store, carry, and forward* paradigm to carry the interest towards the AoI. Thus, to reduce the number of interest bursts, and to avoid collisions in such a network, the broadcast needs to be regulated by each participating device itself. In [8], Amadeo et al. show that a set of timers (called *defer time*) used to schedule interest broadcast on a forwarding device can alleviate this problem. Based on this observation, we also introduce our *defer time* parameters for the two-phase forwarding mechanism.

In the very first step of the two-phase forwarding, the gateway devices after receiving the sensing tasks from the tasking server will assume the role of the *information consumers*, and start broadcasting the interest packets. The *information consumers* include their own distance towards the AoI into the interest packets before broadcasting. Participating devices upon receiving the broadcast interest packet will determine which phase of the *two-phase* they are currently in and process the interest packet accordingly. Regardless of forwarding phases, if data matched with the named interest can be found in the CS of any devices, the data will be forwarded backwards to the consumers directly. To schedule broadcast for the matched data packet at node N_i, the timer DT_{data} is determined as:

$$DT_{data} = Time_{DeferSlot} * \left(\frac{d_{N_i \to}}{d_{max}}\right) + T_{Random} \qquad (3)$$

In the above equation $Time_{DeferSlot}$ is a system configurable parameter which indicates the maximum value of defer time, $d_{N_i \to}$ is the distance from N_i towards the AoI, d_{max} is the maximum distance from a consumer towards the AoI which can be extracted from the interest packet, and T_{Random} is a random component added to avoid data/interest collision. With the defer timer for broadcasting a data packet determined in this way, the defer time for data packets at devices nearer to the AoI will be shorter (since for these devices, component $d_{N_i \rightarrow}$ is smaller). Such a defer timer works in favor of improving the quality of the data, since the data cached near to the AoI is in general fresh collected data, i.e., more relevant according to the time requirements. Furthermore, as another effect of the proposed defer time for data packet, data stored at devices farther from the AoI but closer to an information consumer will have to wait longer before being rebroadcast. This effect increases the chance for such devices to receive more current data, which are forwarded back from the information producer. Hereby, the old data cached on devices nearer to the AoI will be replaced, leading to better data quality in general. Additionally, each device maintains two priority queues for the packets, a *high priority* and a *low priority* queue. The packets in the *high priority* queue will always be scheduled for broadcasting before the packets from the *low priority* queue. To ensure that the requested data can reach the information consumer as soon as possible, data packets are always pushed into the high priority queue, while interest packets are pushed into the *low priority* queue.

In case there is no matched data for an interest at a device, this device will rebroadcast the interest according to the two-phase forwarding mechanism. In the *approach phase*, each device determines its current distance towards the AoI and determines the total number of interest packets which this device has broadcast until the current observation time. These two attributes will be included into an interest packet before it is being rebroadcast. As a result, the interest packet will contain three pieces of information, i.e., (i) the current distance of the corresponding devices before broadcasting the interest packets, (ii) the maximum distance from the *information consumers* to the AoI as observed by the corresponding devices, and (iii) the total number of interest packets that have been broadcast by the corresponding devices up until now, as shown in Figure 7.

Thanks to the embedded information, each device receiving an interest packet can make an autonomous decision on how to further forward the interest. The forwarding/rebroadcasting decision is made based on (i) the current distance, (ii) the moving characteristics (direction, speed), and (iii) the residual energy of the corresponding devices.

To account for the moving direction of a device with regards to the two-phase forwarding concept, we determine the angle Θ between the moving direction of this device ($\overrightarrow{md}$) and the straight direction ($\overrightarrow{N_i}$) from this device towards the center of the AoI. This process is illustrated in Figure 9.

The angle Θ is calculated using the slopes of the two direction vectors as follows:

$$\Theta = \tan^{-1}\left(\frac{|\overrightarrow{md}| - |\overrightarrow{N_i}|}{1 + |\overrightarrow{md}||\overrightarrow{N_i}|}\right) \tag{4}$$

A device only decides to rebroadcast, i.e., to forward the interest packet if rebroadcasting works in favor of bringing the interest packets nearer to the AoI. With regards

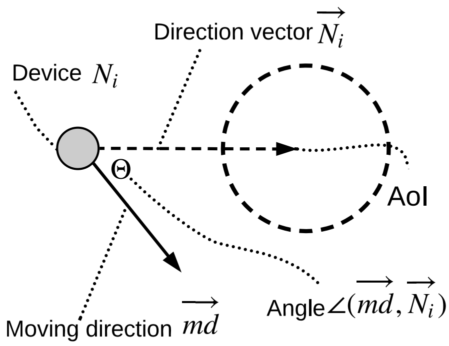


Figure 9: Illustration angle between moving direction of device N_i, and straight direction towards AoI to estimate how far device N_i is moving away from AoI direction.

to distance factor and moving characteristics, a device will drop an interest packet if its current distance towards the AoI exceeds the distance value extracted from the received interest packet, since this indicates that the corresponding device is located farther from the AoI compared to its neighbors. Forwarding interest towards the direction of the AoI is realized through rebroadcasting only from devices located closer to the AoI. Additionally, two thresholds can be set, i.e., a threshold for the moving speed of participating devices and an angle threshold Ψ for the angle Θ between moving direction and straight direction towards AoI of a device as calculated above. Such thresholds only allow the devices that are moving fast towards the *buffer zone* to rebroadcast. Furthermore, stationary devices such as emergency routers are also eligible candidates for rebroadcasting, since such devices are in general connected to power sources and are able to reliably buffer interest packets within the opportunistic networks. With regards to the residual energy, a threshold for energy reading can also be set to make sure that the participating devices should be able to carry the interest packets to the information before running out of energy. All in all, considering distance, moving characteristics, and residual energy ensures, that only the best forwarders will rebroadcast, thus improving forwarding time and reducing overhead.

Next, we elaborate on the defer timer for rebroadcasting interest packets DT_{Int} to counter the interest burst problem in NDN-based opportunistic networks. Our defer timer is calculated as follows:

$$DT_{Int} = Time_{DeferSlot} * (T_d + T_e + T_s + T_{md}) + T_{Random} \tag{5}$$

in which:

$$\begin{aligned} T_d &= w_d * \frac{d_{max} - d_{N_i \rightarrow}}{d_{max}}, \quad T_e = w_e * \frac{e_{max} - e_{N_i}}{e_{max}} \\ T_s &= w_s * \frac{s_{max} - s_{N_i}}{s_{max}}, \quad T_{md} = w_{md} * \frac{\Psi - \Theta}{\Psi} \end{aligned} \tag{6}$$

In Equation 6, each factor of the defer timer for interest packet has a weighting value based on how important each factor should contribute to the forwarding.

T_d, T_e, T_s, T_{md} are the factors related to distance, energy, speed, and moving direction of the corresponding devices, respectively. T_{Random} is a random component included to avoid interest collision. In this way, the defer time at the best forwarding devices will be less, i.e., when devices are moving farther away from *information consumers* (compared with distance d_{max} as the distance from consumers towards AoI, extracted from interest packet), when devices have a high energy level (compared with threshold e_{max}), when devices move fast (compared with threshold s_{max}), and when devices move straight towards AoI (compared with moving angle threshold Ψ). As a result of this defer time, interest packets can be expected to be forwarded towards the AoI as fast as possible.

Next, we elaborate on the *wait phase*. The pseudocode for *wait phase* is given in algorithm 2. The *wait phase* is activated, when the distance $d_{N_i \to} < R_{BZ}$. Devices entering the *wait phase* are devices located within the buffer zone. As such, the objective is to quickly find a mobile producer within the AoI. To this end, the buffer zone in our concept basically represents the *attract* gradient field to attract as well as pull the interest faster towards the AoI. Accordingly, we realize this phase by adapting the defer timer for interest packets. Within the *wait phase*, we replace the factor d_{max}, which was the maximum distance from the information consumers towards the AoI with the size R_{BZ} of the buffer zone radius. Since R_{BZ} is in general smaller than the distance between information consumers and AoI, replacing d_{max} with R_{BZ} will make the defer timer for broadcasting an interest shorter. Thereby, the chances to find the appropriate mobile producers close to the AoI are increased. As soon as a device enters the AoI, the factor d_{max} will be further replaced by R_{AoI}, which is even less than R_{BZ} aiming to increase the chance to reach mobile producers even more when being inside the AoI. In NDN-based opportunistic networks, an interest packet is rebroadcast only when receiving an incoming interest. We therefore introduce the concept of replicated (re)broadcasting for the *wait phase*. Through replicated (re)broadcasting, the devices in the *wait phase* can schedule rebroadcast of an interest packet multiple times without having to rely on an incoming interest packet. The number n_{REP} of replicated interest packets is calculated as follows:

$$n_{REP} = n_{max} * \frac{R_{BZ} - d_{N_i \to}}{R_{BZ}} \tag{7}$$

in which, n_{max} is the configurable maximum number of replication. This value can be determined or adjusted based on the available residual energy at each device. Calculating the number of replicated interest packets in this way will increase the number of replication when a device moves closer towards the AoI, potentially leading to higher chances to find mobile information producer.

Our two-phase forwarding mechanism is designed to achieve short *time to find mobile information producer*, reducing overhead and interests congestion for the participating devices. As discussed in Section 3.1, forwarding/broadcasting in an uncoordinated way can result in wastage and degradation of network performance. The reason is due to the energy consumption of participating devices caused by each redundant broadcasting, possible collisions, sensing etc. Since we consider an opportunistic network as the communication medium for emergency response, this issue has to be addressed. We deal with this problem indirectly by addressing the fairness for the forwarding process. For this purpose, each device includes the *total number of broadcast packets* as an attribute into the interest packet before broadcasting. This value also serves as shared context information for other devices to make their broadcasting decision. Accordingly, each device after receiving several interest packet will determine the median value (m_b) of the broadcasting numbers, extracted from the overheard interest packets. Thereby, the device can compare its current number of broadcasting interest packet b_i against m_b. Having $b_i < m_b$ suggests, that the corresponding device has contributed less in forwarding process than its neighbors. As a result, this device will decrease its current defer time $DT = DT * \frac{b_i}{m_b}$. Through this adjustment, an underutilized device will (re)broadcast more frequently regardless of which phase it currently is in. Since this adjustment is executed each time a new interest packet is broadcast in the network, the participating devices adapt themselves over time with respect to their neighbors, leading to more efficient resource consumption overall.

3.4 EVALUATION

With the design of the context-aware two-phase forwarding protocol, we focus on the distribution of sensing tasks in an opportunistic network, considering the emergency response scenario. Therefore, the evaluation for this chapter especially focuses on how our interest forwarding concept can cope with the requirements for crowd sensing as well as with the harsh conditions of the emergency response situation. Next, we will elaborate on the configuration of our evaluation setup. We divide the evaluation results into three categories, i.e., time related metrics, communication overhead, and fairness measure. Altogether, these metrics allow us to analyze the performance of the forwarding concept w.r.t. crowd sensing requirements as well as study the respective trade-offs.

3.4.1 *Evaluation Setup*

We relied on the NDNSim network simulator [1] to implement and evaluate our forwarding concept. NDNSim is an NS-3 [68] based network simulator. Thus, all network models available in NS-3 can be reused. NDNSim implements the fundamental designs of the NDN paradigm architecture as proposed originally in [217]. While on packet level, NDNSim is compatible with the implementation of NDN which is used in testbed and real devices, using NDNSim for evaluation allows us to generate large scale simulative experiments. Furthermore, since NDNSim is based on NS-3, we can make use of the existing interface to incorporate mobility of devices into the evaluation. NDNSim provides the abstraction of *face* to realize an NDN-based application. To create an application on a mobile opportunistic ad hoc network, the *NetDeviceFace* as the abstraction for communication on the link layer is provided. The *defer timers* can be incorporated into the *NetDeviceFace* directly for the purpose of reducing interest bursts, and avoiding interest collision. The rest of the forwarding concept (drop, rebroadcast) can be realized as *forwarding strategy* with the application abstraction *AppFace*.

As the baseline used to compare with our approach, we implemented three other forwarding mechanisms: *geo-forwarding*, *controlled-flooding*, and *pure flooding*. *Geo-forwarding* is implemented as geocast for NDN-based opportunistic ad hoc networks regarding the predefined AoI. In general, implementing geocast for opportunistic ad hoc networks is to forward the packets to the neighbors with a better chance to reach the destination [112], using context information such as the current distance towards AoI, the frequency of devices visiting the AoI etc. In an emergency response scenario, the environment tends to be affected which leads to rapidly changed mobility and chaotic behavior of human-carried devices in large-scale scenario [204]. Due to this reason, we leverage only the current distance towards the AoI of mobile devices for realizing *geo-forwarding*. As such, after receiving an interest packet, only mobile devices with shorter distance towards the AoI will be eligible to rebroadcast. This behavior can be realized based on our implementation of the two-phase forwarding concept by disabling the *wait phase*. In this way, *geo-forwarding* in our evaluation still follows the behavior of geocast for opportunistic networks, while also benefitting from the consideration of energy, moving characteristics in the *approach phase* of our concept. However, geo-forwarding alone is not designed to counter the mobility of *information producers*.

Controlled flooding is based on the implementation of Amadeo et al. [8]. Controlled flooding relies only on the defer timer with random factor to avoid interest collision. Besides defer timer and broadcasting, there is no other special mechanism affecting the forwarding behavior. As such, controlled flooding can serve as a good baseline in this case, similarly to epidemic flooding as in normal opportunistic networks [147, 207]. Lastly, *pure flooding* relies on the default behavior of the NDN paradigm, in which for each incoming interest the participating devices will rebroadcast immediately without waiting for defer time. Hence, the *pure flooding* is used not only for comparison, but also for analyzing the need of adjustment in order for the NDN paradigm to function properly on opportunistic ad hoc networks.

We set up a simulation scenario to represent a crowd sensing application in an emergency response situation, in which the sensing tasks are distributed from the *information producers*, in form of interest packets, through the opportunistic ad hoc network. According to this scenario, the mobile devices simulated for the evaluation can communicate with each other through the 802.11 WiFi model. We use the *YansWiFiChannel* model of the NS-3 simulator, which leverages the 802.11g WiFi model and Rayleigh propagation model [177]. Correspondingly, we set the $\mathtt{Time}_{\mathtt{DeferSlot}}$ to be a multiple of 28 μs. The base 28 is set according to the Distributed Inter-Frame Space (DIFS) for the 802.11 WiFi family [15]. We base our work on and extend the patch for NDN *NetDeviceFace* developed by Amadeo et al. [8]. Thereby, interest rebroadcasting is enabled, consequently, multiple-hop transmission over an opportunistic ad hoc network with NDN paradigm is achieved.

The structure of the simulation scenario is illustrated in Figure 10. We simulate an area of size $800 \times 800\ m^2$. The AoI has a radius of 50 m, and is located near the bottom of the simulated area. We create 12 *information consumers*, located near the top of the

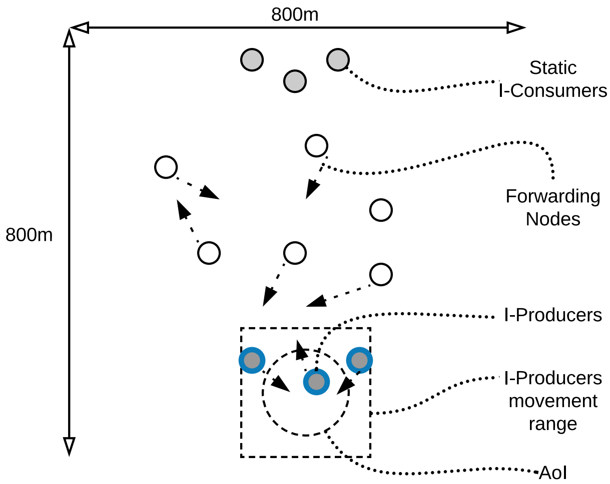


Figure 10: Illustration of the simulation scenario for evaluation of decentralized crowd sensing task distribution.

simulated area. For the evaluation in this chapter, we concentrate more on the performance of the interest forwarding concept, therefore, we set the *information consumers* to be static. We will evaluate the *mobile information consumers* later together with the concept of results delivery in Chapter 5. The *information producers* are mobile and located around the AoI. We set up a movement range surrounding the AoI, in which *producers* can move freely and randomly. As a result of this set up, the *information producers* will be available inside the AoI arbitrarily, thus simulating the unavailability of *information producers*. Note that, an *information producer* provides requested data only when it is located inside the AoI. For our scenario, we created 25 *information producers*. We generated and varied the number of *forwarding nodes* (from 20 to 100 nodes), which are the devices used for rebroadcasting the interest packets between the location of the static consumers and the location of the AoI. The forwarding nodes are also mobile. To challenge our concept, we decided on using the *RandomWalk2dMobilityModel* for this evaluation, since we wanted to first simulate the possible uncertain behavior of humans within the first hours of a disaster relief scenario, i.e., to assess the *chaotic* and *dynamic* interaction among devices [144]. Each forwarding device moves with a movement speed between 2 and 5 m/s, which approximates the running speed of a pedestrian [224]. We equip the forwarding devices with two energy consumers, i.e., the *WifiRadioEnergyModel* and the *BasicEnergySource*. The *BasicEnergySource* model will reduce the energy unit of the simulated devices over time, while the *WifiRadioEnergyModel* will reduce the energy unit of the simulated devices when they communicate via WiFi interface, i.e., every time a device broadcasts an interest. To incorporate heterogeneity into the scenario, we assigned an arbitrary energy level between minimum 3000 and maximum 19000 Joules to the forwarding devices according the normal distribution. (Note that, 19000 Joules corresponds to 1300 mAh, which is the typical capacity of a smart phone). We choose the energy threshold of 10000 Joules, i.e, only

Table 3: Simulation parameters for the evaluation of crowd sensing tasks distribution

Parameter	Value
Simulated area	$800 \times 800\,\mathrm{m}^2$
Number of forwarding nodes	20, 40, 60, 80, 100
AoI radius	50 m
Transmission range	100 m
Energy capacity	3000 − 19000 Joules
Velocity of nodes	2–5 m/s
Mobility model	`RandomWalk2dMobilityModel`
Energy model	`WifiRadioEnergyModel`, `BasicEnergySource`
Buffer zone radius	100 m, **150 m**, 200 m, 250 m, 300 m
Simulation time	18-20 hours

devices having more than half of their battery left will rebroadcast. We install the NDN-based ad hoc protocol stack on all simulated devices. Table 3 summarizes the most important parameters for our simulation scenario.

We varied the configuration parameters, i.e., the number of forwarding devices (from 20 to 100) and the size of the buffer zone radius (from 100m to 300m) for the simulation as summarized in Table 3. For each configuration, we repeated the run 100 times with different random seeds to obtain more dependable results. The obtained results can be divided into three categories, i.e., time related metrics, overhead metrics, and fairness metrics. For time related metrics, we determined the *time to find producer* and the *end to end delay*. *Time to find producer* is calculated as the time elapsed after the first interest packet is broadcast from the *information consumers* until the first *information producer* receives the interest packet while being inside the AoI. *End to end delay* is calculated as the time elapsed after the first interest packet is broadcast until the first data packet is received at one of the *information consumer*. While the performance of the interest forwarding mechanisms is mainly characterized by *time to find producer* metric, the *end to end delay* can consider the performance for data retrieval as a whole by incorporating both *interest forwarding phase* and *data forwarding phase* as results delivery process. With regards to the overhead measurement, we determine the total number of interest broadcasts by all participating devices and the total summed energy consumption of all participating devices. These two metrics characterize the trade-off that has to be made for each forwarding mechanism. Last but not least, we measure the fairness of the forwarding devices with regards to their contribution to the forwarding mechanism. We use the well-know Jain's fairness index [23] to quantify the fairness factor. Jain's fairness index is calculated as:

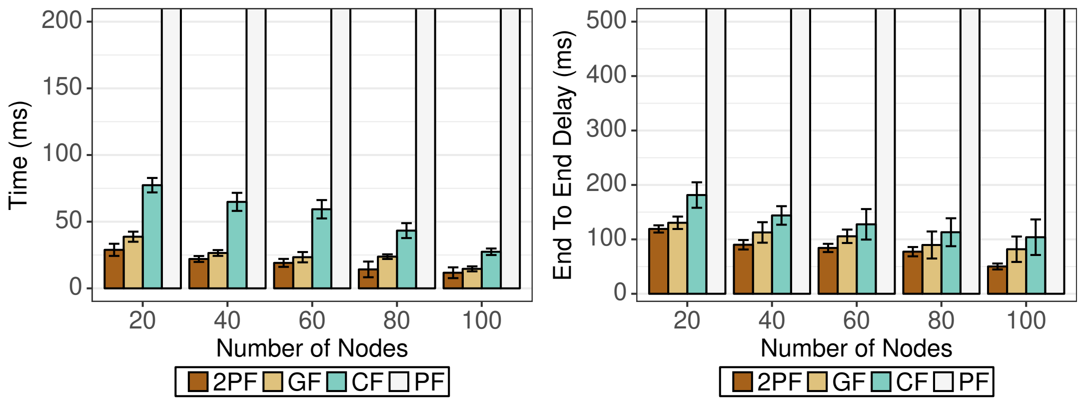
$$JI(x_1, ..x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n * \sum_{i=1}^{n} x_i^2} \tag{8}$$

This index is calculated over $x_1, .., x_n$ as the measurement for resource consumption on the considered entities $e_1, .., e_n$ of a distributed system. For our purpose, we use the number of broadcast interest packets as the indicator of resource consumption on forwarding devices. The Jain's fairness index yields a value between $[0, 1]$, which can be translated to the equality level of resource consumption, distributed on the percentage value of the considered entities. Hence, a value closer to 1 means better fairness for the whole system.

Having elaborated on the setup of the evaluation scenario and on the evaluation metrics, in the next sections we will discuss the obtained results.

3.4.2 *Time related Metrics*

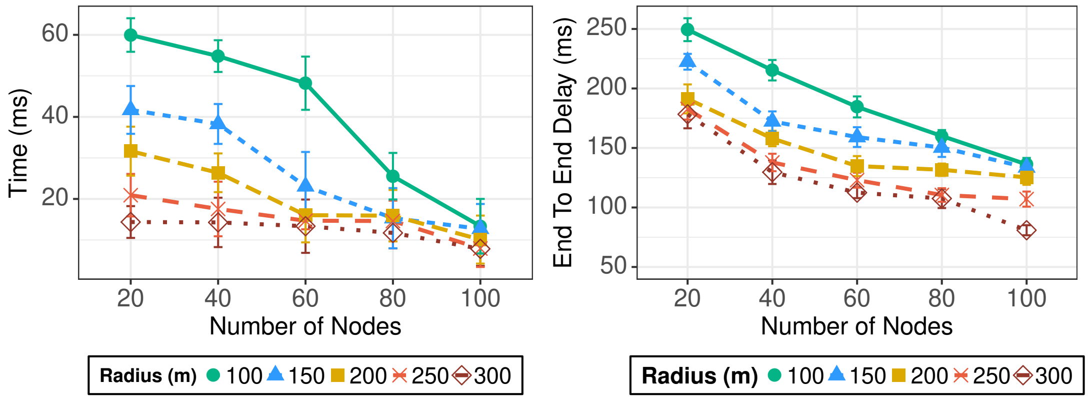
We measured *time to find producer* and *end to end delay*, to (i) compare the performance of our two-phase forwarding concept against the other three approaches *geo-forwarding, controlled flooding, and pure flooding* with varying network density and (ii) to assess the effect of the *buffer zone* notion introduced in our concept, since the optimal size of the *buffer zone* cannot be determined in advance in the considered scenario. Figure 11 presents the results obtained for (i), while Figure 12 presents the results obtained for (ii). In all result plots, we use abbreviations to name the forwarding approaches. *2PF* represents our two-phase forwarding concept, *GF* represents the geo-forwarding, *CF* represents controlled flooding, and lastly, *PF* represents pure flooding concept.



(a) Time to find producer (b) Data delivery time

Figure 11: Time comparison among interest forwarding approaches with varying number of forwarding devices.

From the results shown for *time to find producer* in Figure 11a, it is obvious that *pure flooding* performs worst of all forwarding approaches. We cut the limit of the time axis to 200 ms, since in all cases, *pure flooding* requires more than 3000 ms to find the mobile information producer. This result is expected, since *pure flooding* assumes the default behavior of directly rebroadcasting an incoming interest, which is designed for the NDN paradigm being applied in a stable network, e.g., wired networks. As such, chances are high that interest broadcasting with pure flooding generates a large number of collisions, degrading the overall performance of the network. This clearly underpins the need to adapt NDN for mobile opportunistic ad hoc networks. By introducing the defer timer for interest rebroadcast as *controlled flooding*, the *time to find producer* can be improved greatly compared to the *pure flooding*. For lower number of forwarding devices, it requires more time to find the mobile producers (with 20 forwarding devices, around 77 ms are needed). A higher network density helps to reduce *time to find producer*, since more forwarding nodes will increase the chance to build-up a path towards the AoI faster. With 100 forwarding devices, the *time to find producer* can be cut down to average 26 ms. The linear increase of the number of forwarding devices also indicates the linear decreasing in the time to find producer. Compared to *controlled flooding*, both *geo-forwarding* and *two-phase forwarding* are able to improve the time to find mobile information producers. The improvement is more obvious with low number of forwarding nodes. In the simulation with 20 forwarding nodes, two-phase forwarding requires on average 28 ms, while geo-forwarding requires on average 38 ms to complete the search for mobile information producer. Hereby, with low network density the two-phase forwarding approach is able to performs better, it requires around 25% less time compared to the geo-forwarding in case of 20 nodes. In general, for all network densities used for the evaluation, two-phase forwarding always performs better than other forwarding approaches with regards to the *time to find producer*. The main reason is that two-phase forwarding with the *wait phase* floats the interest request around the AoI waiting for mobile information producers to show up inside the AoI. In other approaches, the interest might reach mobile information producers when these are located outside the AoI, thus making the producer unsuitable to provide data. The performance gap among forwarding approaches is linearly decreasing with increasing number of forwarding nodes. For instance, with 100 forwarding nodes two-phase forwarding requires on average 17 ms, while geo-forwarding requires around 23 ms and controlled flooding requires around 33 ms. This observation suggests that the two-phase forwarding might be more suitable and be able to offer more benefit for low density opportunistic networks. Figure 11b shows the end to end delay as the data delivery time for the information consumers. Likewise to the evaluation for time to find producers, pure flooding also performs worst in this case, confirming again the unsuitability of such concept in both interest and data forwarding. In all network densities, two-phase forwarding and geo-forwarding allow for faster data delivery time for information consumers compared to controlled flooding. Two-phase forwarding is still able to perform better than geo-forwarding and controlled flooding with all network densities. The end to end delay, as the time required to deliver data depends on both interest forwarding and data forwarding.



(a) Time to find producer

(b) Data delivery time

Figure 12: Assessment for the effect of the radius size of the buffer zone for two-phase forwarding with varying number of forwarding devices.

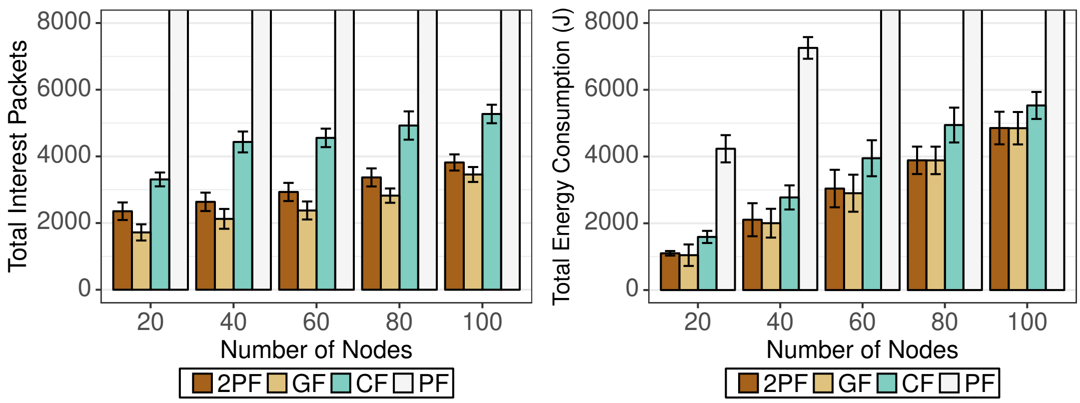
By providing faster *time to find producer*, the two-phase interest forwarding already contributes to improving the overall data delivery time as a whole.

Previously, we have raised the issue of our concept that it would be impossible to determine an optimal value for the *buffer zone* of the wait phase. While replicated rebroadcasting in the buffer zone can float the interest packet around the AoI to wait for mobile information producers thus improving the time to find a producer (cf. Figure 11), this will come with a trade-off for overhead which will be discussed later. Therefore, we analyze how the radius size of the buffer zone affects the time related performance metrics. Figure 12 presents the results when varying the size of the *buffer zone* in our two-phase forwarding approach. For the AoI with radius size of 50 m, we vary the radius size of the buffer zone from 100 m to 300 m. Figure 12b presents the end to end data delivery time when varying the size of the buffer zone radius. This result shows that, regardless of buffer zone radius end to end data delivery time decreases with increasing number of forwarding nodes. Such a result is expected, since from the previous evaluation results we could learn that 100 forwarding devices are sufficient to cover the simulated area, leading to faster forwarding time. Interestingly, with regards to the time to find a producer, the obtained results confirm that there is indeed a correlation between the size of the buffer zone radius and this metric. We can observe in Figure 12a, that there is a greater time performance gap among varying buffer zone radii with sparse network density of 20 forwarding nodes. In this case, a buffer zone of 300 m only requires on average 17 ms, while a buffer zone of 100 m increases the time by a factor of 3, i.e., 60 ms to find mobile information producers. The reason for this performance time gap with different radii size can be explained, by the fact that in a sparse network forwarding devices might have fewer neighbors, thus the choice for a suitable forwarding device (e.g., closer to the AoI, having enough energy, etc.) might not always be possible. As a result, a larger buffer zone radius allows the forwarding devices to float the interest not only to wait and reach mobile producers faster, but also to wait for a suitable forwarding device to appear as a new neighbor.

The performance gap among different buffer zone radii is smaller with increasing number of forwarding nodes in the network. With 100 forwarding nodes, the time to find producers for all radii almost converges. The performance gap among buffer zone radii is marginal. From this insight, the size of the buffer zone radius can be configured in an adaptive manner to balance both the performance and the trade-off overhead. For a sparse network larger buffer zone radius size is required while for a dense network only a small buffer zone will be sufficient.

3.4.3 *Overhead Assessment*

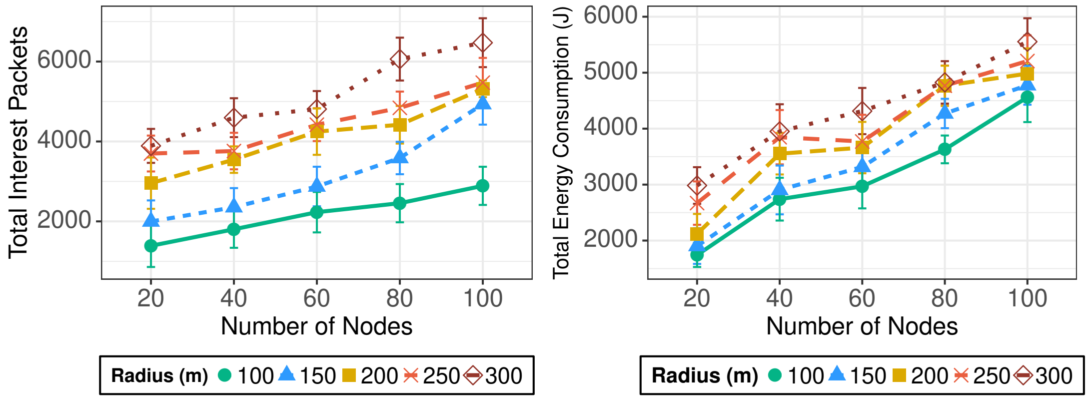
In the previous section, we have analyzed and compared the performance of our two-phase forwarding approach against our selected benchmarks. The results have confirmed that our two-phase forwarding mechanism performs better than other approaches with regards to searching for mobile producers to provide data from the requested AoI. By design, the two-phase forwarding mechanism has to take an overhead trade-off into consideration (by introducing replicated rebroadcast) in order to find mobile producers faster. To assess the overhead, we measured the total number of interest packets which are broadcast in the NDN based opportunistic network and the overall energy consumption of all devices. The main source for energy drain in the simulation is when a device broadcasts, thus higher energy consumption means more overhead for the forwarding approach. The results can be observed in Figure 13. Figure 13a shows the total number of interest packets after the simulation for the forwarding approaches with varying number of nodes, while Figure 13b shows the total energy consumption. Figure 14 shows the overhead with regards to varying buffer zone radii.



(a) Total number of broadcast interest packets (b) Total energy consumption of all nodes

Figure 13: Overhead Comparison among Interest Forwarding Approaches.

Expectedly, when increasing the size of the buffer zone radius both total number of broadcast interest packets and total energy consumption increase linearly. This again pleads for a careful choice of the buffer zone radius as discussed previously in



(a) Total number of broadcast interest packets

(b) Total energy consumption of all nodes

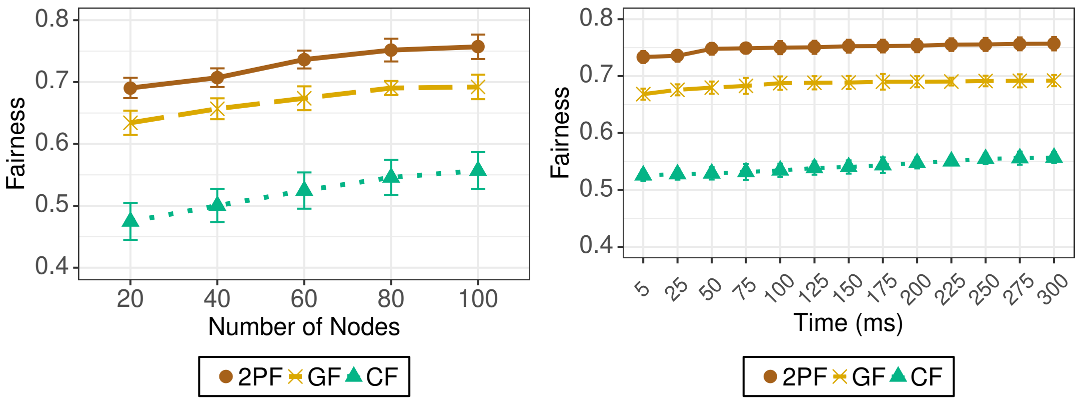
Figure 14: Overhead generated when varying buffer zone radius in the two-phase forwarding approach with varying number of forwarding devices.

order to balance between the overall performance and the generated overhead of two-phase forwarding concept. In Figure 13, it can be observed that the overhead caused by pure flooding far exceeds the other approaches. Two-phase forwarding and geo-forwarding are able to reduce overhead compared to controlled-flooding in terms of both generated interest packets as well as energy consumption. Due to the introduction of replicated rebroadcasting inside the buffer zone, the overhead generated by two-phase forwarding is more than geo-forwarding, which deactivates the *wait phase* from the two-phase forwarding concept. However, in comparison to the controlled flooding two-phase forwarding still generates less overhead for all network densities. With 100 forwarding nodes, two-phase forwarding generates on average 3800 interest packets, while controlled flooding generates more than 5200 interest packets. With regards to resource consumption, two-phase forwarding and geo-forwarding are comparable. This suggests, that despite generating more interest packets, two-phase forwarding mechanism is still able to reduce the energy consumption. This effect can be explained by the fact, that we introduce a mechanism to regulate fairness for forwarding among devices. We study this effect in the following section.

3.4.4 *Fairness regarding Forwarding Contribution*

To analyze the fairness with regards to forwarding contribution, we determined Jain's fairness index as shown in Equation 8 over the total number of interest packets which are broadcast at each forwarding device during the simulation. Thereby, we compared the fairness index of our two-phase forwarding approach against the geo-forwarding and controlled flooding (pure flooding is omitted due to its inefficiency). We obtained the results for the fairness index with varying number of forwarding nodes and the fairness index evolution over time; both of which are shown in Figure 15.

Regardless of time and network density, our two-phase forwarding approach was able to provide a better fairness index compared to both geo-forwarding and controlled



(a) Fairness index w.r.t. varying number of nodes

(b) Fairness index w.r.t. time

Figure 15: Fairness index w.r.t. forwarding contribution

flooding. The best fairness index can be achieved with 100 forwarding nodes for all approaches. In this case, the two-phase forwarding mechanism yields a fairness index value of around 0.75, while this value is on average 0.7 and 0.55 for geo-forwarding and controlled flooding respectively. The geo-forwarding as implemented in the simulation basically disables the wait-phase; as such, it benefits from the same regulation mechanism to regulate the defer time within the approach phase, which explains the high fairness index of 0.7. The better fairness index of two-phase forwarding is due to the fact, that through two phases the interest packets are forwarded more directly towards the AoI. Figure 15a shows a dependency of fairness index values on the number of forwarding nodes. Fairness index values increase linearly with increasing number of forwarding nodes. The reason is, with a higher number of forwarding nodes the network is denser. In a denser network, more devices can receive the broadcast interest packets, which allow them to extract the context attributes embedded in each packet to self-regulate the defer time. Figure 15b shows that the fairness index in all approaches also increases linearly with increasing simulation time. However, the increasing trend is not as noticeable as compared to the trend when varying the number of forwarding nodes. Overall, the results confirm the effect of a self-regulation mechanism, introduced in our concept to improve fairness in forwarding for the whole network.

3.5 DISCUSSION

In this chapter, we introduced our concept to distribute sensing tasks in mobile opportunistic networks based on the NDN paradigm. To this end, we proposed (i) a hybrid communication architecture, (ii) a naming scheme for interest packets, which captures the quality requirements of crowd sensing application while enabling the user to specify the granularity level of accuracy for the requested data, (iii) a modified construction of interest packets, which embeds context information to allow for distributed coordination of participating devices, and (iv) a two-phase forwarding mechanism which successfully distributes the sensing tasks to the mobile information producers in a decentralized manner. Through the evaluation, we assessed and showed that our forwarding approach is able to utilize the opportunistic resources and the mobility of forwarding devices, while introducing only minimum overhead. Thereby, the distributed coordination for the forwarding only relies on local shared context information embedded in each interest packet. This confirms the feasibility of opportunistic resources utilization without centralized coordination.

Sensing task distribution accomplishes the first objective of our research goal, to trigger data collection based on crowd sensing for a decentralized scenario such as a disaster relief situation. Thus, the evaluation presented here focuses more on this aspect. Even though, we have presented results for data delivery to information consumers, we leave a deeper analysis and an extension for data delivery in Chapter 5. The data, as successfully collected through two-phase interest forwarding concept, are still raw measurements. To extract valuable information from these data, one option is aggregate all data at the gateway devices and offload to a central cloud server for further processing. However, in several situations of emergency response, the decision has to be taken timely and locally. Thus, centralized solution for processing data is not suitable. The alternative solution is to process data directly within the network, leveraging idle computing resources of mobile devices. The distributed in-network processing approach will be addressed in the next chapter. Finally, due to the focus on interest forwarding we evaluated our concept with static information consumers. In general, information consumers can also be mobile, e.g., first responders in emergency situation might distribute a sensing task to collect information along the way, while moving to other locations. In NDN-based networks, the problem of moving information consumers is considered to be solved by letting the information consumers reissue their interest request, which will be served by matching data cached within the network [198]. However, caching data and propagating data all over the network also consume resources, which are scarce in emergency situations. Hence, to enhance the data delivery, we later propose and evaluate the integration of mobility prediction for the data forwarding phase. Overall, the mechanism to distribute crowd sensing tasks presented in this chapter provides high quality measurements for *information retrieval* as a whole.

4 DISTRIBUTED PROCESSING

In the previous chapter, we have discussed on how to distribute sensing tasks to trigger data collection in a decentralized manner, utilizing mobile devices as data sources. To extract valuable information from the collected data, these data have to be processed. In case the communication infrastructure is available, an obvious choice for processing crowd collected data is to upload data to a central cloud server for aggregation and sophisticated data analysis, as a cloud server can provide sufficient resource for data storage as well as data processing. According to Kumar et al., such computation offloading can help to save energy compared to processing data directly on mobile devices [83]. This practice is common for the crowd sensing paradigm [54, 74, 129]. However, this turns the data processing cloud server into a single point of failure. Several drawbacks of this model are noticeable. First, the privacy of the participants for such centralized data collection is a growing concern [67], since the cloud server can theoretically extract other information of the participants not withstanding just the information intended for crowd sensing campaign. Second, the centralized model is not applicable when the access to the cloud server is impossible due to either the failure of server itself or due to the impaired communication infrastructure, e.g., in disaster situations. As a consequence, there is a shift against computation offloading to a cloud. As alternative solutions for computation offloading to cloud servers, other locally offloading models are possible, e.g., offloading to nearby edge devices such as cloudlet-upgraded routers [122, 123], or leveraging distributed computing resources of local mobile devices for processing [41]. Due to the fact that the process to extract information from data tends to be a complex computing task which involves several processing stages and requires different logic operations [134], offloading computation on multiple mobile devices for distributed processing is more favorable.

In this chapter, we propose a model to enable leveraging idle computing resources of participating mobile devices in an opportunistic network to process and extract valuable information from crowd collected data directly within the network. Again, we tailor our model to work in a decentralized fashion, which relies on distributed coordination and avoids any centralized entity. In an emergency response scenario, such model is well-suited to facilitate relief operations; since first responders can use the extracted information to organize the local relief works efficiently. For example, a complex image processing task, which requires the execution of several resource-intensive operations can be divided into several tasks and executed by participating devices in order to determine the number of victims in an emergency situation [133]. Before going into details of our design, we first discuss the challenges and requirements for a distributed processing concept in Section 4.1. The details of our solution to enable distributed data processing will be given in Section 4.2. We present the models to describe the complex operations and to allow for distributed coordination respectively in Section 4.3 and 4.4. Based on our solution for enabling distributed processing, we further propose several *local strategies* for participating devices presented in Section 4.5, aiming to increase the quality of the overall distributed processing. Our solution is an enabling technique for distributed processing in mobile opportunistic network in general and to facilitate data processing in crowd collected data in particular. Therefore, in order to focus on the analysis and evaluation of our distributed processing concept as a whole, we create a customized simulation that allows us to concentrate on the *processing aspect* of the concept. Later, in Chapter 5, we will show how our distributed processing concept can be integrated into the information retrieval workflow.

4.1 REQUIREMENTS AND CHALLENGES

Our solution introduced in this chapter is based on the fact that modern mobile devices are capable of executing complex operations despite the fact energy consumptions on these devices remain a constraint [66, 172]. Thereby, the idle resources of mobile devices owned by the participants can be volunteered to create a distributed processing environment [189]. The notion of *volunteer computing* holds true especially in disaster situations, in which the information extracted from the distributed data processing can be used to offer emergency services, which are beneficial for all participants (this form of incentive has been discussed in Section 3.1 for general crowd sensing applications). Other form of incentive is the reservation of resources in disaster situations. Lieser et al. [94] propose a resources market, in which the users can compete and reserve energy resources for recharging. With regards to *volunteer computing*, a resource market can be extended so that, the devices which provide more computing resources will be able to reserve for more energy recharging. Overall, in the considered emergency response scenario, we assume that all participating devices are willing to contribute their computing resources. Due to the dependence of the distributed processing concept on mobile devices as well as on participants as human carriers of these devices, the following challenges for distributed processing have to be addressed:

- *Uncertainty*: One of the biggest challenge for providing distributed processing in this setup is the uncertainty as a result of a rapidly changing *opportunistic* network. Since mobile devices as processing units are carried by human participants, the quality of the processing highly depends on the uncontrolled mobility of the participants. The uncertainty caused by human mobility can for instance result in the unavailability of a special operation or a special hardware that is required to complete a complex processing task. Uncertainty can also lead to an incorrect decision when assigning processing tasks, which affects the overall performance.

- *Resource constraints*: Even though modern mobile devices are capable of processing complex operation, we still have to consider them as resource constraint. This is due to the fact that (i) a mobile device might have to execute several tasks, e.g., sensor reading, processing at the same time and (ii) recharging mobile devices in emergency situations might be difficult.

- *Heterogeneity*: As discussed in Section 3.1, participating mobile devices possess different capabilities. With regards to distributed processing, two types of heterogeneity can influence the overall performance, i.e., (i) heterogeneity in capabilities and (ii) heterogeneity in available resources. Heterogeneous capabilities, similar to *uncertainty*, imply that not all devices are capable of executing a particular operation required for the computing tasks. Heterogeneous sources imply that the participating devices might have different energy level left or possess different Central Processing Unit (CPU), which makes them perform differently in distributed processing.

Having discussed the challenges, the QoS requirements for distributed processing in highly dynamic environments such as mobile opportunistic networks can be derived and classified into two categories, i.e., functional and non-functional requirements. The QoS parameters are derived from relevant research work in *mobile cloud computing* [11, 53, 189] and in-network processing paradigm [199].

Functional Requirements:

Due to the uncertainty challenge, the most important goal for a distributed processing concept is to ensure the correctness for the processing, i.e., the outcome of the distributed processing has to be as intended. The QoS parameter, which implies this aspect, is the *success rate*. Together with the uncertainly, it is challenging to complete a complex processing task with multi-stages operations. Hence, a high success rate can capture how well a distributed processing concept can cope with the uncertainty of the environment.

With regards to the integration with a crowd sensing application, the goal of distributed processing is to extract information from crowd collected data. As such, the quality of the processing will impact the quality of the information. In addition to the success of a complex processing task, the time factor also plays an important role; since delay in extracting information can make this information less relevant or even not usable under the circumstances [124]. As a result, low *completion time* of the distributed processing concept is the second requirement.

Overall, fast and successful processing of complex tasks cover the functional requirements, which are mainly derived from the uncertainty of the considered environment.

Non-Functional Requirements:

The non-functional QoS requirements are derived taking into consideration the issue of resource constraints of the participating mobile devices. The energy consumption of a mobile device depends on the execution code and the functions called on this device [66]. As a result, we cannot influence the energy consumption of the distributed processing itself. Nevertheless, we need to make the enabling concept *communication and computation efficient*, i.e., the resource consumption for communication and computation overhead required for distributed coordination need to be minimum.

Additionally, *computation/load balancing* is another goal desired as non-functional requirements for distributed processing [91, 128]. Thereby, the *resource constraint* is addressed by dividing and distributing the portions of the complex processing task equally on the participating devices. Thereby, the load is distributed. Hence, the resource consumption will also be shared among devices equally. This can also potentially lead to an improvement of the overall performance of the distributed processing. Furthermore, since the distributed processing concept relies on the participation of humans, fair computation/load balancing might also be another incentive to motivate participants to offer their idle computing resource.

4.2 THE ADAPTIVE TASK-ORIENTED MESSAGE TEMPLATE — ATMT

In this section, we first present the *adaptive task-oriented message template* proposed to enable distributed processing in a mobile opportunistic network. The goal of the *task message* is to allow the participating devices to cooperate with each other through distributed coordination. To cope with the problem of uncertainty caused by the network, a distributed coordination scheme has to be fast and precise. From an architecture point of view, one way to enable distributed coordination without relying on a centralized coordination entity is to distribute the coordination itself to multiple coordination entities. In dynamic mobile networks the coordination entities can be chosen, e.g., by clustering the devices and assigning the coordination function to the corresponding cluster head [31, 171]. Hereby, each cluster keeps track of which computing services, which resource are available in its cluster in order to assign the distributed processing task. However, in this approach, within each cluster the devices still have to rely on a single cluster head for coordination, which might not react fast enough with uncertainty. Hence, in order to cope with uncertainty, in our approach we enable autonomous decision of each participating device. Each participating device decides for itself how much it should contribute and what it should do next for the distributed processing. As such, our solution follows a fully-decentralized approach. We rely on shared context information of each device to facilitate distributed coordination as well as autonomous decision of individual devices. Inspired by the idea to enhance cloud monitoring in [137] where the monitoring tasks are defined and coordinated through a template, in our mechanism, we propose a message template to define the processing tasks for distributed processing.

Our enabling concept for distributed processing is named *adaptive task-oriented message template* (abbr. *ATMT*), which basically provides a construction template for users to define a complex processing task as the final goal, its corresponding operations as well as the execution order of these operations to accomplish the processing goal. The *task messages* will be distributed into an opportunistic network formed by mobile devices for distributed processing. Hereby, each device, when receiving the *task message* can make autonomous decision on how to proceed by itself. The autonomous decision is possible by bundling and embedding relevant context information of the processing task as well as the payload data required for the processing altogether in a *task message*. As such, each *task message* is a *self-encapsulated* message which describes itself. The illustration of the message template is shown in Figure 16.

We divide an *ATMT* message into the *ATMT* header which contains context information and the *ATMT payload* which contains the data required for the processing task. To enable continuous processing, the *ATMT payload* can also contain the results

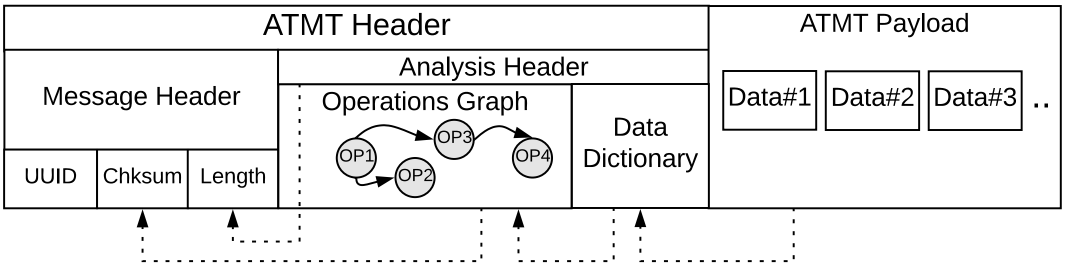


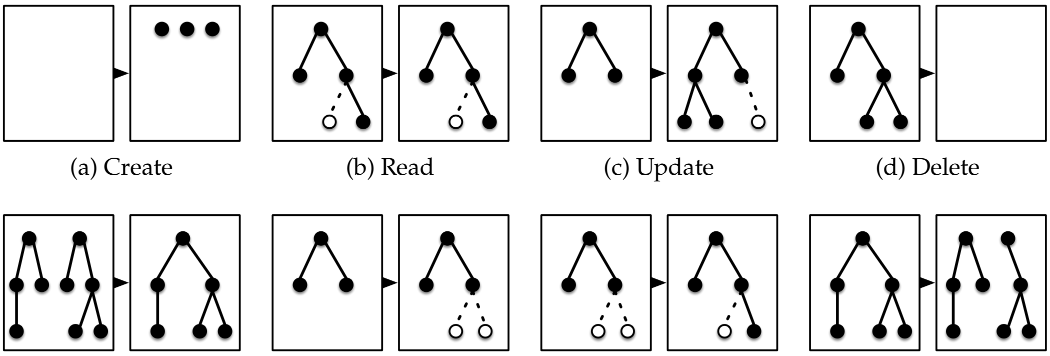
Figure 16: Adaptive Task-oriented Message Template (ATMT) for enabling distributed in-network processing (image from our publication in [136]).

of successfully executed operations. As such, the *ATMT* payload is mixed and can have varied lengths depending on the types and the total number of unexecuted operations. Therefore, in order to make each *ATMT* message communication efficient, we support compression of the data pieces required for the operations so that the total size of each *ATMT* message remains small enough for efficient communication in a mobile opportunistic network. To allow for more flexibility, each data piece associated with one particular operation can be compressed using different encoding methods. The association of data and operation is achieved through the *ATMT* header. Within the *ATMT* header, we further divide the header into a *message header* and an *analysis header*. The *analysis header*, as its name suggests, contains information required for *data analysis* as processing task. Two sub-components are required for the *analysis header*, i.e., an *operations graph* and a *data dictionary*. As aforementioned, we need a mechanism to associate the payload data, which are compressed with different encoding with the corresponding operations. This association is realized through the *data dictionary*. Basically, the *data dictionary* is a data structure to map data to operations. It also indicates start and end of the payload data. The *operations graph* defines the operations required to complete the computing task and the processing order of the operations.

Within the *ATMT message header*, we include an Universally Unique Identifier (UUID), a *checksum*, and a *length* field. The *length* field indicates the total length of the analysis header, since the size of the *operations graph* can also vary for different computing tasks. With the length of the *analysis header* given in the *message header*, a device is able to distinguish between the *ATMT header* and the *ATMT payload*. This information is useful if a device wants to check the context information without having to read and parse the content of the payload. Thereby, the UUID and the *checksum* can be used. Since an *ATMT* task message template can be duplicated in a mobile opportunistic network, when a device receives several task messages, it can check and drop messages with the same UUID. As a result, messages with different UUID will be considered differently. The *checksum* is a field that represents the current state of the processing task. If the *checksums* of two messages are identical, it indicates that the processing state of both messages are the same and one message can be dropped. We only determine the *checksum* over the *operation graphs* due to two reasons: (i) knowing the processing state is sufficient for a participating device to decide whether or not to participating in the processing or to drop the messages, and (ii) due to the autonomous characteristic of our concept and the heterogeneity of the devices, two devices processing the same operation might result in slightly different results (e.g., if these two devices use different execution codes libraries).

4.3 GRAPH-BASED PRIMITIVES

We dedicate this section to elaborate on the *operations graph* introduced as one essential field of the *analysis header* for the *ATMT* messages. To distribute the computation, it is common to divide an intensive computing task into several operations. Therefore, a complex computing task contained in an *ATMT* message can consist of more than one operation. The goal of the *operations graph* is thus to represent the goal of the processing task, the corresponding operations, and the execution order of the operations. For this purpose, we use a Directed Acylic Graph (DAG) to contain all of the required information.



(a) Create (b) Read (c) Update (d) Delete

(e) Merge (f) Delegate (g) Operate (h) Split

Figure 17: Illustration of graph-based primitives for *ATMT* messages (image from our publication in [136]).

The *operations graph* of an *ATMT* message is thus modeled as a graph $G_O = (V_O, E_O)$; with V_O being the set of vertices and E_O being the set of directed edges. The set V_O consists of vertices with each vertex indicating an operation. Each directed edge in the set E_O, which connects two vertices as two operations, indicates the dependent execution order between these two operations. For instance, edge $V_x \rightarrow V_y$ implies that the operation represented by vertex V_x has to be executed before the operation contained in the vertex V_y and that the output of the executed operation in V_x is required as an input for operation in the vertex V_y. The resulting *operations graph* is topological sorted to ensure that overall the graph-based representation for the execution oder of the operations lead to the final processing goal. Therefore a device reading the *operations graph* can traverse the graph quickly to look for an operation that can be executed by the respective device.

We formally define several *graph-based primitives*, which provide us an interface to interact and to *operate* on the *ATMT* messages. The illustration of examples for all *graph-based primitives* can be found in Figure 17. Four essential *graph-based primitives* are required: *create* (C), *read* (R), *update* (U), and *delete* (D) which are abbreviated as *CRUD* primitives. Additionally, we define four *sub-primitives* for the *update* (U) primitive. These are named *merge* (U_M), *operate* (U_O), *delegate* (U_D), and *split* (U_S), which are abbreviated as *MODS* primitives. Table 4 and 5 summarize all defined *graph-based* primitives as an overview.

Table 4: Common primitives on the *ATMT messages* (*CRUD*)

Abbr.	Names	Meaning
C	**(C)**reate	Create a new *ATMT* message, with unprocessed, raw payload data attached
R	**(R)**ead	Parse the message and read its content
U	**(U)**pdate	Adjust the content of the message, i.e., *operations graph* and payload data can be modified (details in Table 5)
D	**(D)**elete	Drop and thus remove the message from the network

Table 5: Possible update primitives on the *ATMT messages* (*MODS*)

Abbr.	Names	Meaning
U_M	**(M)**erge	Merge and join two *ATMT messages* into one, i.e., merge the operations graph and the corresponding data
U_O	**(O)**perate	Execute an operation on the corresponding attached data
U_D	**(D)**elegate	Construct a new *operation graphs*, or add a new operation to the existing *operations graph*
U_S	**(S)**plit	Split an *ATMT* messages into two different ones

We now elaborate on how the *graph-based primitives* should be used with the *ATMT* concept. In a crowd sensing application, a device after collecting data can *create* an ATMT message and attaches the collected raw data with the created message. Within the *operations graph* the raw data are associated with the root node of the graph to signify the begin of the execution order. Devices in the network can *read* the whole content of an *ATMT* message. Note that the *read* primitive does not imply any modification on the content. For the purpose of modifying the content of an *ATMT* message, the *update* primitive is used. There are four types of modifications possible for the *update* primitive. (i) The *merge* primitive is used to merge the operations graphs and contents of two *ATMT* messages. The *merge* primitive is defined, as the chances are high that *ATMT* messages are duplicated and processed differently in a mobile opportunistic network. (ii) The *operate* primitive is used when a device possesses an operation required next in the execution order of the current ATMT message and decides to execute this operation. The *operate* primitive also allows the processing device to replace the raw payload data with the result obtained from the executed operation. (iii) The *delegate* primitive is used to enable a device to construct a new *operations graph* and to associate the raw payload data with the operations. Devices that are allowed to carry out a *delegate* operation are those with domain knowledge of how to process and analyze the collected data. (iv) The *split* primitive is used to divide a single *ATMT* messages into two. If there are different ways (e.g., using different operations) to obtain the same goal, the current *ATMT* task can be duplicated by dividing and representing using two different (sub-)operations graph, which increases the chance of success for this task. Finally, the *delete* primitive can be used by a device to drop and thus remove an *ATMT* message from the network.

Overall, the set of graph-based primitives as defined allows participating devices to act autonomously on receiving *ATMT* messages. In the next section, we will elaborate on how the autonomous action of each device contributes to the distributed processing as a whole.

4.4 SYSTEM MODEL FOR DISTRIBUTED PROCESSING

To utilize *ATMT* messages for enabling distributed processing, we rely on the autonomous decision and the heterogeneous capabilities of the participating devices. In a nutshell, the participating devices are able to cooperate with each other in a decentralized self-organizing manner. As such, the distributed coordination and cooperation among participating devices are realized through four different roles, i.e., *sensor*, *operator*, *forwarder,* and *delegator*. These roles specify the capabilities of the participating devices. The capabilities and the descriptions of the roles are summarized in Table 6.

Table 6: Roles and capabilities of participating devices as defined in the distributed processing model using *ATMT* messages

Role	**Capabilities**	**Description**
(S)ensor	{C}	Initiate *ATMT* messages with raw payload data attached
(O)perator	$\{U_O, U_D\}$	Execute one or more operations on the message's data, and update the result within the *ATMT* payload
(F)orwarder	{R}	Read and forward *ATMT* message unchanged
(D)elegator	$\{U_M, U_D, U_S, D\}$	Adjust the content *ATMT* message, via merge, delegate, split, delete

A device assuming the *sensor* role has to be able to obtain data through built-in sensors. After obtaining data, a *sensor* device can initiate the distributed processing work-flow by *creating* an *ATMT* message and attaching the obtained data. In a mobile

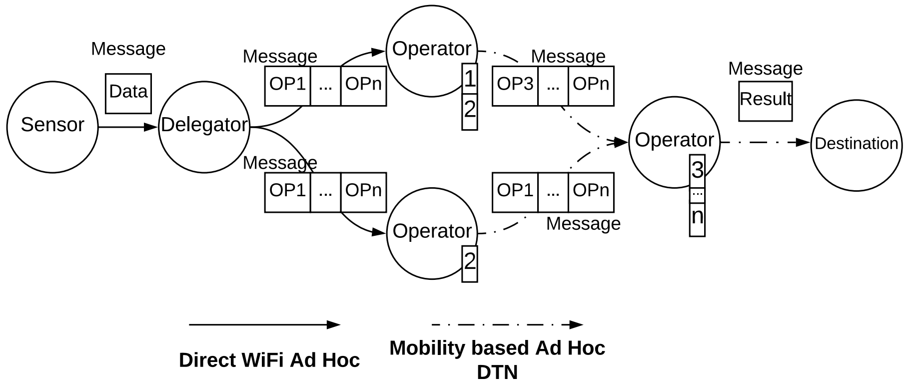


Figure 18: Illustration of distributed processing workflow using *ATMT* concept.

opportunistic network, the communication relies on the *store, carry, and forward* approach, in which messages are forwarded from one device to another in order to reach the destination at a later time. Accordingly, each device with networking capabilities in our system model can assume the *forwarder* role. A *forwarder* role is equipped with *read* capability, so that each device can parse and read content of an *ATMT* message in order to make further decision such as merge or drop. The participating devices that decide to contribute computing resource for the distributed processing, assume the role of *operator*. *Operator* devices, upon receiving an *ATMT* message, can read the content of the *message header* and the *analysis header* to determine whether this device possesses the operations/services that can be applied on the upcoming operation of the *operations graph*. If an *operator* device decides to perform an operation from an *ATMT* message, this *operator* will replace the raw data input from the payload of the *ATMT* message with the output of the executed operation using the *update* primitive. Unsolicited payload data can also be completely removed from the *ATMT* payload. As a result, after several iterations, in which operations from the *operations graph* are executed, the size of the *ATMT* payload can be decreased thus reducing the generated network traffic. This mechanism contributes to making the communication of the distributed processing concept more efficient. Last but not least, the core of the distributed coordination is the *delegator*. *Delegators* are devices with domain knowledge to construct an *operations graph*. The *delegators* lead the distributed coordination for the processing by adjusting the content of *ATMT* messages when necessary. As such, a *delegator* device can perform all *update* primitives on *ATMT* messages, i.e., *merge*, *split*, *delegate*, or *delete*. A common workflow of *ATMT* enabled application, involving every defined roles, is illustrated in Figure 18.

In our system model, the roles can be assigned or assumed by the participating devices dynamically based on their available capabilities, on current utilization, or on events triggered from the applications [163]. As a result, a device might have several roles at the same time based on its capabilities. With regards to security, the roles with critical functions such as *delegator* can be assigned in a static manner. Since *a delegator* should also possess the domain knowledge on how to process the data, the devices capable of *delegator* role normally belong to an authorized organization. For instance, in emergency response situations, such devices might be devices from authorized first responders or firefighters. In case, multiple sensors are required, the data need to be collected and aggregated at a predefined location before being attached with processing information by the *delegators*. Further mechanisms to establish a decentralized trusted environment for distributed processing such as in [44] can be leveraged.

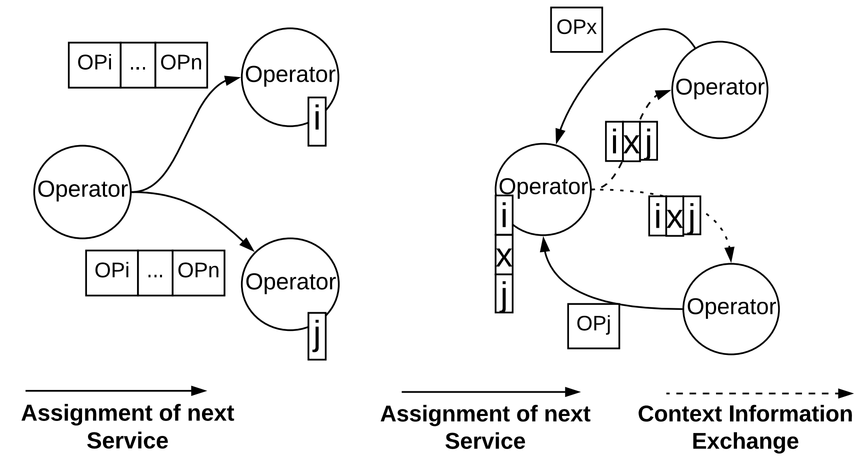
To match the operations for execution, *operators* can compare the name or the representation indicated within the *operations graph*. To reduce communication and computation overhead, there is no need for an *operator* to employ ontology to derive knowledge of the processing. Since most of the functions and operations on each devices should be defined during design time, a simple comparison for matching is sufficient. To represent the names of the operations, Internationalized Resource Identifiers (IRI) can be used or an application-agnostic naming scheme similar to the NDN paradigm can be leveraged [59].

In the distributed processing workflow shown in Figure 18, the *ATMT* message template is the enabling concept and the forwarding of an *ATMT* message from one device to another is the main driver for distributed processing. It is obvious, that a single device cannot handle the whole workflow, which requires numerous capabilities. Each device thus operates on a part of the *ATMT* task and forwards the result after the operation to other devices via *store, carry and forward* networking paradigm for further processing. We define the act of passing a processed *ATMT* message on to other devices as *task handover*. Through *task handover*, the success rate for completing a complex computing task can be increased. However, if not designed properly, the handover can generate much overhead. Instead of increasing the success rate, poorly designed task handover can even decrease the overall performance. In the next section, we propose several handover strategies for *ATMT* messages, aiming to satisfy the *QoS* requirements while considering the challenges resulting from mobile opportunistic networks as discussed in Section 4.1.

4.5 OPPORTUNISTIC TASK HANDOVER

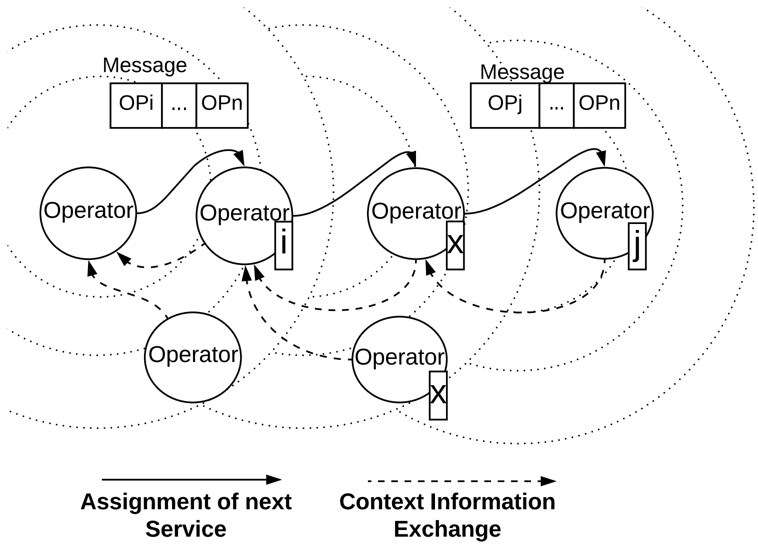
In this section, we present several task handover mechanisms for our proposed *ATMT* concept. This section is named *opportunistic task handover*, due to the fact that the communication among participating devices mainly depends on opportunistic ad hoc WiFi. In this environment, two devices can exchange information and handover *ATMT* task messages upon each opportunistic contact.

Recapitulating from Section 4.1, we design mechanisms for task handover that aim to improve the *QoS* parameters, satisfying both functional and non-functional requirements. Consequently, the task handover mechanisms should achieve high success rate, low completion time, generate minimum overhead, and be able to achieve computation/load balancing among participating devices. As a basis to derive the handover mechanisms, we rely on the framework proposed by Alakeel et al [6], which is a general guideline to achieve load balancing while also considering other performance aspects of a distributed system. According to the authors, a decision to assign a task to



(a) Naive task handover based on flooding

(b) Work-stealing based task handover



(c) Task handover based on local optimization

Figure 19: Illustration of ATMT task handover mechanisms.

an entity of a distributed system needs to consider three main aspects, i.e., *information strategy, transfer strategy*, and *location strategy*. *Location strategy* refers to the set of rules which determine where to assign the task. With regards to our processing mechanism, location strategy refers to which destination should we offload/handover an *ATMT* message. *Transfer strategy* refers to the set of rules which determine *whether* to perform a task locally on the corresponding entity or to offload the task to others. *Information strategy* refers to the set of rules, which determine how to obtain the context information in order to generate the rules for *location strategy* and *transfer strategy*. As a result, the *information strategy* plays a central role for designing handover mechanisms. With regards to mobile opportunistic networks, the context information can be obtained at different granularities: (i) only local context information of individual devices is considered, (ii) devices can share their current status with others. Hereby, the granularity level can be further distinguished between shared context information with one-hop direct neighbors or shared context information with devices being multiple hops away. Based on the granularity level of shared context information as observed, we propose three strategies for *ATMT* task handover designed for mobile opportunistic networks. These are named *naive*, *work-stealing*, and *local optimization*. According to the granularity level of shared context information, in *naive* mechanism, a device only leverages its own context information and status to make handover decisions. The *work-stealing* strategy leverages shared context information of one-hop direct neighbors, while *local optimization* utilizes shared context information of multiple-hop away devices. As sharing context information generates communication overhead, each strategy has its pros and cons. In the following sections, we provide details of the three proposed handover mechanisms. The proposed handover mechanisms are illustrated in Figure 19.

4.5.1 *Naive*

The *naive* mechanism as its name suggests, is a simple handover mechanism, which is based completely on the context information of a single device. These are available services/operations, the current processing workload, as well as the resource utilization on the corresponding device. We model the resource utilization on each device through a queue of *ATMT* task messages. Thus, the total resource which can be made available and contributed to the distributed processing, is represented by the total length of the *tasks queue*, i.e., the maximum number of *ATMT* task messages, that a device is willing to take in. Having a task queue and its limited size, two strategies are conceivable for *naive* task handover: (i) *greedy* full-capacity handover, and (ii) *limited* capacity handover.

In our considered scenario, a processing goal in an *ATMT* message is a complex task, which is divided into multiple operations, that need to be invoked and executed in a particular order to complete. As a consequence, the longer the *operations graph*, the more difficult and challenging it is to successfully carry out the operations, since not all operations are available on all devices. To counter this issue, the *greedy* handover requires all participating *operator* devices to contribute as much as possible, i.e., the whole capacity of their *task queue* should be utilized. After performing the operations from *ATMT* tasks, a device will handover all the processed *ATMT* messages to all of its direct one-hop neighbors for further processing. Thereby, each *ATMT* task is duplicated and carried on multiple devices, which is intended to increase the success chance for a processing task. The behavior of greedy handover is inspired by the concept of *epidemic flooding* in opportunistic networks [200]. Thus, greedy handover also shares the advantages as well as the disadvantages of *epidemic flooding*. The greedy handover strategy makes a trade-off of communication and computation overhead for higher success rate by generating a high number of duplicated *ATMT* messages in the network. Similar to *epidemic flooding*, the greedy handover can serve as a baseline for the performance comparison among the task handover strategies.

Taking into account the main drawback of the greedy handover strategy, in limited capacity handover, we aim to balance between redundancy overhead and performance of the naive mechanism. Under the observation, that several mobile devices might possess similar capabilities to perform an operation in a dense network with high number of devices, it is possible to limit the resource contributed by the participating devices while not negatively affecting the overall success rate. The success rate can be compensated among devices with similar capabilities. Thus, a device following a limited capacity strategy can put a threshold on its task queue. As soon as the number of received *ATMT* message reaches the threshold, a device will reject/drop further messages. Limiting the task queue might also benefit the overall computation balancing, since each device ensures that it is not over-utilized. We will assess this intuition later in the evaluation.

In summary, for the naive task handover mechanism, a device in our system follows two strategies as previously discussed: (i) full offering of available resources on *operator* devices thus resulting in an *epidemic flooding* of *ATMT* tasks in the whole network and (ii) limiting the size of acceptable *ATMT* tasks in the queue and dropping further incoming messages.

4.5.2 *Work Stealing*

Our second handover mechanism is named *work-stealing*. *Work-stealing* is a notion defined in the early age of parallel computing [16] and refers to the act of re-allocating threads among processors. In parallel computing consisting of multiple processors, when a processor notices that it is current running only a few threads, indicating that this process is under-utilized, this under-utilized process can take over threads from other over-utilized processors. As a result, the over-utilized process is relieved, leading to improved performance of the whole system. This idea is later generalized in the context of mobile crowd computing [52], in which *work-stealing* refers to the act that a mobile device can take over/steal computation tasks from the other. Compared to the concept proposed by Fernando et al. in [52], our work-stealing concept differs in the sense, that we do not rely on centralized coordination of any dedicated devices. Instead, each mobile device in our system can act autonomously. Accordingly, each *operator* in our system is capable of stealing processing jobs from other *operators*. In work-stealing, each operator constantly checks on the current number of *ATMT* messages in its tasks queue. If this number is smaller than a predefined lower-bound threshold, this *operator* considers itself as being under-utilized. Thereafter, an under-utilized *operator* will send a *work-stealing* message to its one-hop direct neighbors to inform the nearby *operator*, that it still has available capacity and is willing to take over some *ATMT* tasks. We assume a trusted environment, which is established among participating devices [44]; consequently, only trusted devices are allowed to take over ATMT tasks in order to mitigate security risks in such decentralized environment.

To regulate the number of *ATMT* tasks, which the *work-stealing* operator is able to accept, we use a parameter n_{ws} as the maximum work-stealing capacity. A *work-stealing* message thus contains the list of operations which can be executed by the *work-stealing operator*, and the maximum number of the acceptable capacity n_{ws}. Since the capacity of the *ATMT* tasks queue on each *operator* device is limited, the work-stealing capacity n_{ws} should not exceed the maximum size of the tasks queue. Additionally, to account for computation balancing in general and to govern the behavior of the operators in order to avoid egoistic handover, we introduce a second parameter n_{keep}. When an *operator* receives a *work-stealing* message, which indicates the maximum number of n_{ws} that this *operator* device is allowed to handover, this *operator* device should always hold back a minimum number of n_{keep} messages in its own tasks queue. Overall, the number of *ATMT* tasks $n_{w-handover}$ being handed over to a work-stealing device satisfies the following conditions:

$$n_{keep} + n_{w-handover} \leqslant size(TasksQueue) \tag{9}$$

$$n_{w-handover} \leqslant n_{ws} \tag{10}$$

To determine the number of *ATMT* tasks $n_{w-handover}$ for handover considering computation balancing, three strategies for work-stealing are conceivable. (i) Each device receiving the work-stealing notification tries to exhaust the full capacity as indicated in the work-stealing message. (ii) The work-stealing device coordinates and regulates the work-stealing process, i.e., the work-stealing device divides the allowed capacity equally to the total number of its direct neighbors. (iii) The work-stealing devices bases the acceptance of task handover on the first come first serve principle. The neighbor, which replies first to a work-stealing message, is allowed to send first with its desired number of tasks which needs to satisfy the aforementioned conditions. As soon as the work-stealing threshold is reached, the work-stealing devices will stop receiving further messages and notify their neighbors on this decision. Thereafter, the neighboring devices will also refrain from handing over *ATMT* tasks.

4.5.3 *Local Optimization*

In this section, we present the third handover mechanism, which is also our main contribution to achieve the QoS requirements as discussed in Section 4.1. The name *local optimization* refers to the fact, that we let each operator perform an estimation locally to determine which is the best candidate for the next handover destination. The handover based on *local optimization* is inspired by an observation of Eager et al. [46]. The authors state that, an adaptation to improve the performance of local components in a distributed system consequently improves the overall performance. Similar observation is confirmed in mobile opportunistic networks by Sadid et al. [167]. Thereby, the authors show that enabling hop-by-hop services composition in opportunistic networks can improve the success rate as well as allow for load balancing. Furthermore, the performance of the hop-by-hop services composition is comparable to the performance of a centralized coordinated services composition. Following this line of thought, we devise our strategies for task handover based on local optimization. Thereby, we only leverage local shared context information of the participating devices to make handover decision. For local optimization, the devices share their status and context information with both their direct one-hop neighbors as well as with devices being multiple hops away. Our goal is to extend the search range for a handover destination, which is beyond the range of the direct neighbors in order to increase the chance to find (more) suitable candidates.

Handover based on local optimization depends on the quality of context information shared by participating devices. In general, there are two options to acquire context information. First, when a device requires information for handover, it sends out a query and waits for other devices to reply with the requested context information. Second, two devices upon opportunistic contact can share with each other their context information in a proactive manner. With respect to mobile opportunistic networks with uncertainty caused by mobility of participating devices, the later approach with proactive context sharing offers more advantages. In an opportunistic network, reactive reply upon a request for context information can return after considerable delay, adding even more uncertainty for the *information strategy*. We employ proactive context sharing for our handover strategy based on local optimization.

Upon any opportunistic contact, devices share and store context information from each other. As a result of storing context information, each device maintains a list of devices together with their current status that the corresponding device has discovered. To enhance the *information strategy* for task handover, each device not only shares its own context information but also shares its list of discovered devices. Exchanging the lists of discovered devices also allows a device to extend or to adjust its own list. The list of discovered devices contains the information required to make handover decisions. The required parameters are listed in Table 7 as follows:

Table 7: Parameters used for task handover based on local optimization

Parameter	**Meaning**
$(op_i..op_j)$	the list of available operations on an operator.
n_u	the resource capacity which is currently utilized by an operator, expressed by the number of remaining tasks in the queue of this operator.
(loc_x, loc_y)	the current location of the corresponding device, expressed by the Cartesian coordinates.
$\vec{v}$	the current moving direction of the corresponding device.
t_{info}	the time stamp generated when the status of the device is captured.

The local optimization is executed by individual devices, which decide to handover a processing task. The local optimization is possible, since each device now possesses a list of discovered candidates, which can be chosen as the handover destination. To counter for uncertainty caused by mobility of devices in an opportunistic network, before triggering the task handover, a device limits the search range for the candidates. Thereby, a device sorts out the candidates, which have the current distance to the corresponding device more than a predefined threshold d_{max}. As a result, only devices located within a proximity of radius d_{max} possessing the next operation required in an ATMT task are considered. Furthermore, the list of potential candidates can be refined by omitting devices that stay around the limit of a predefined search range d_{max} and their moving direction $\vec{v}$ is outwards from the search zone. This sorting is due to the fact, that these candidates might not be available anymore when the handover decision is made. From the eligible candidates, we derive a function to estimate the cost for handover to each candidate. This cost value allows us to rank the candidates and to choose the candidate with minimum cost as the handover destination. The cost function for an operator O selected as a handover destination is calculated based on the current utilization n_u of the tasks queue on the candidate and a corresponding uncertainty factor $\mu(N_O)$ as:

$$c(N_A, N_O, \#_{OP}) = (w_l * c_l * \#_{OP} + w_d * c_d) * \mu(N_O) \tag{11}$$

in which:

$$\begin{aligned} \mu(N_O) &= 1 + \frac{t_{current} - t_{info}}{t_{keepAlive}} \\ c_l(N_O) &= \frac{n_{max} - n_u}{n_{max}} \\ c_d(N_A, N_O) &= \frac{d(N_A, N_O)}{d_{max}} \end{aligned} \tag{12}$$

In Equation 11, N_A denotes the handover triggering device. N_O is a potential handover destination according to the snapshot network view of device N_A. $\#_{OP}$ indicates the number of operations that are chosen for handover. w_l and w_d denote the weights of cost values for load c_l and distance c_d parameters of the handover candidate N_O.

Equation 12 shows the details on how the uncertainty factor $\mu(N_O)$, the costs estimation c_l for distance, and load c_d are calculated. We explicitly consider load and distance for the cost estimation of a handover destination candidate, since these are the two main factors that affect all other QoS requirements (i.e., success rate, completion time, load balancing, overhead). The uncertainty factor is calculated based on the time stamp t_{info}, at which the context information of candidate N_O is generated and the current time stamp $t_{current}$, when the cost function is being calculated. Consequently, the uncertainty factor depends on the elapsed time between t_{info} and $t_{current}$, since more elapsed time also means that the context information available at the current time might get outdated. For instance, the handover destination candidate N_O might possess different state at the current time due to mobility or due to interaction with other operator devices. In this manner, the cost calculated for outdated information results in higher cost value, suggesting a negative handover decision. Thus, the handover triggering device can avoid candidates with outdated information based on higher cost values. The cost estimation for load component is determined based on the current utilization of the tasks queue n_u on the candidate device and the maximum size n_{max} of the tasks queue. $c_l(N_O)$ thus estimates the available capacity of the candidate device to accept more *ATMT* tasks. Accordingly, the handover destination candidate with highly-utilized task queue will get a higher cost, since choosing such candidates as handover destination will increase the chance of offloaded tasks being dropped, when reaching the destination at its full capacity. Last but not least, the distance component of the cost function is determined as the ratio of the distance between the triggering devices and the candidate devices, and the search range d_{max}. The cost estimation for distance factor is calculated this way to prefer handover destinations in closer proximity, since a farther handover destination might require more hops to offload the *ATMT* tasks, leading to more communication overhead. As a consequence, the configurable parameter d_{max} for search range should be chosen carefully to balance between quality of the handover and generated overheads. Similar to the *buffer zone* concept introduced in Chapter 3, an optimum search range in such decentralized fashion is not possible, however, the search range can be adjusted based on the density of the network. In the evaluation, we evaluate the effect of the search range on the handover based on local optimization.

Having the cost function to compare and choose the handover destination candidates among each other, we require a condition to trigger the handover process. We use the load as the indicator to trigger handover. This is due to the fact that when an operator device is overloaded, it can potentially decrease the overall performance of the distributed processing. Further incoming tasks at an overloaded device will be dropped and currently running tasks cannot be completed on time under over-utilized situation. As a result, two triggering conditions are possible. (i) *Proactive* triggering handover occurs every time the shared context information, i.e., the list of discovered devices is updated. With new information, the triggering devices might find better candidates as handover destinations, thus offloading tasks to these candidates can improve the QoS parameters. (ii) *Reactive* triggering handover occurs every time an operator device notices an over-utilized situation (e.g., when the task queue is full or when a predefined threshold is exceeded). As previously discussed, over-utilized devices can impact all QoS parameters. Triggering handover based on load is therefore valid. Using task handover based on local optimization, it is also possible to optimize the energy consumption on participating operators. For instance, if the delivery time does not have to be guaranteed, then the operators which do not lie on the optimal data delivery path can be chosen for task handover; since these devices are in general not over-utilized for communication.

All in all, the task handover based on local optimization always ensures that the cost to offload *ATMT* tasks to a handover destination stays minimum, therefore guarantees that after each handover decision, the best handover candidate for the next offloaded operation is chosen, subsequently improving the overall QoS parameters.

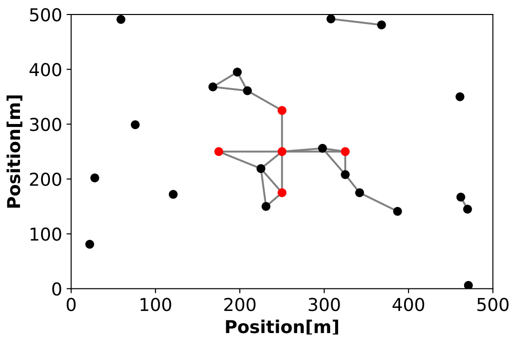
4.6 EVALUATION

With the concept of the *adaptive task-oriented message template—ATMT*, we have designed a solution to enable distributed processing in opportunistic networks in general and to allow for distributed analysis of crowd collected data in particular. Therefore, in this chapter we evaluate the *ATMT* concept and the proposed handover mechanisms from an application-layer perspective, since our concept can be utilized for a wide range of applications which rely on distributed processing. To this end, we implement the *ATMT* concept and the handover mechanisms, using a customized simulator based on OMNeT++ [201]. We focus on the QoS parameters for distributed processing on mobile opportunistic ad hoc networks and how the proposed handover mechanisms effect these QoS parameters. Next, we discuss our evaluation methodology, i.e., the setup and the evaluation metrics. As the first step of the evaluation, we analyze the handover mechanisms individually to find out the advantages and disadvantages of each mechanism. Based on the obtained results of the analysis of individual mechanism, we compare the performance of all proposed handover mechanisms with their best configurations against each other to find out the trade-off between performance and overhead.

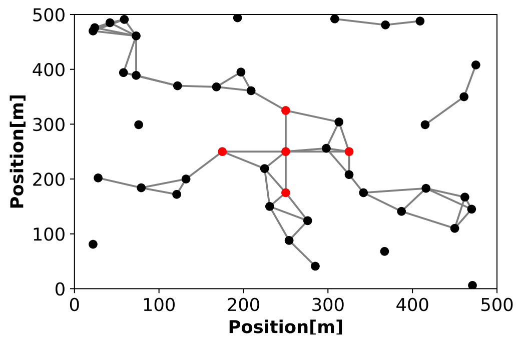
4.6.1 *Evaluation Setup and Evaluation Metrics*

Since the main goal of the evaluation is to analyze the concept of *ATMT*, and the proposed handover mechanisms with regards to distributed processing from an application layer perspective, we implement an OMNeT++ module, that is compatible to the design of the *ATMT* template [134], and on the strategy module that realizes the handover mechanisms [136]. The OMNeT++ simulator provides a complete protocol stack, that allows users to plugin and use the WiFi ad hoc communication interface all from the network layer down to the physical layer. However, simulating with the whole stack generates a lot of overhead and restricts the scalability of the simulation scenario. Therefore, we abstract the WiFi ad hoc communication module, so that when two devices are in WiFi range of each other they will be able to communicate and exchange messages. We assume, that the congestion control and collision avoidance of the network layer can be handled using access control mechanisms proposed for the link layer such as in [104]. Due to our focus on the application layer and handover strategies, such abstraction does not significantly affect the results of the evaluation.

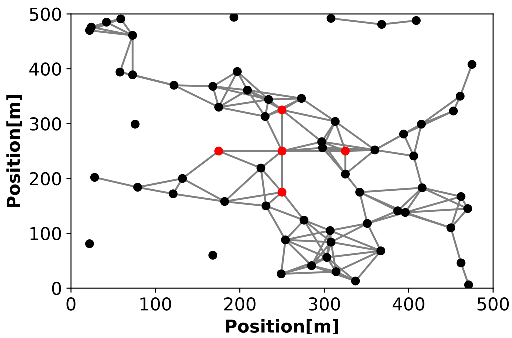
Since we design *ATMT* targeting mobile opportunistic networks, we create a simulation scenario which allows us to generate and disseminate *ATMT* based computing tasks in such a network. We simulate an area of $500 \times 500\ \mathrm{m}^2$, in which mobile devices as operators can move around. To inject and disseminate the computing tasks into the simulated area, we use five static delegator nodes, one main delegator, and four helper delegators, located around the main delegator. This setup can be observed in Figure 20. The five red points represent the static positions of the delegators. Complex computing tasks are generated by the main delegator, which first replicates these tasks on the helper delegators. When a mobile device appears in communication range with one of



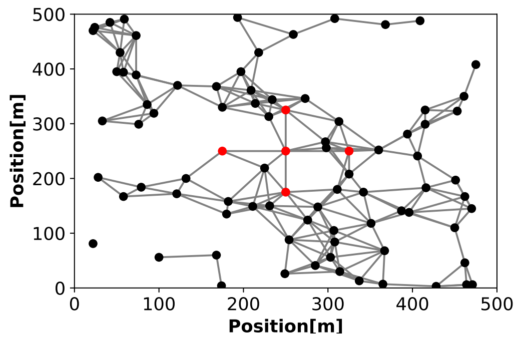
(a) 20 Nodes



(b) 40 Nodes

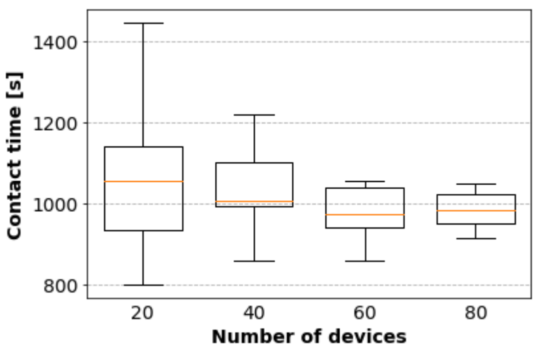


(c) 60 Nodes

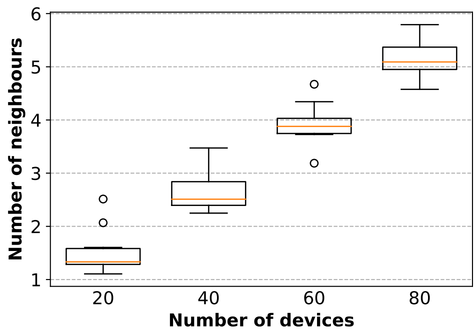


(d) 80 Nodes

Figure 20: Initial distribution of participating mobile devices and positions of delegators in the simulation scenario.



(a) Contact duration



(b) Number of neighbors

Figure 21: Analysis of contact durations and number of neighbors in the simulated scenario.

the helper delegators, this mobile device will receive the computing tasks. Thereby, the computing tasks are successfully injected into the opportunistic network. The reason behind this setup is to allow the generated computing tasks to reach multiple mobile operators even if the network is sparse as shown in Figure 20a with only 20 devices.

Table 8: Simulation parameters for evaluation of ATMT handover mechanisms

Parameter	**Value(s)**
Simulated Area	$500 \times 500\,\mathrm{m}^2$
Simulation Time	1 hour
#Nodes	20, 40, 60, 80
WiFi Range	75 m
Mobility Model	`LevyWalkMobilityModel`
#ATMT-Tasks	100, 1000
Naive	full, limited
Work Stealing	full, FCFS, equalized
Local Optimization	proactive, reactive

Note that, this setup is intended to generate a "fair" starting phase for all simulation scenarios, in which tasks are successfully injected to multiple operators in the network regardless of network density. This setup however does not affect the performance of the handover mechanisms w.r.t. QoS parameters, since the performance of the distributed processing mainly relies on the mobility, the interaction, and the behavior of participating devices during the simulation.

To assess the performance of the handover mechanisms with regards to the complexity of the computing tasks, we create two types of tasks: (i) *simple* tasks which require between two and three operations to complete the processing goal, and (ii) *complex* tasks which always contain five operations. Each operation is represented by a predefined name and a predefined execution time which is required on the participating devices, that possess the services to execute this operation. The mobility model plays an important role in the evaluation, since the QoS parameters for distributed processing in this case depend on the mobility of participating devices. According to Rhee et al. [160], the Levy walk mobility model exhibits relevant characteristics of human mobility in a disaster relief situation. Based on this observation, we use the Levy walk mobility traces, generated using BonnMotion [10] on all participating devices in the simulation. Figure 21 shows the characteristics of the generated mobility traces used in our evaluation with regards to average contact time between nodes and average number of neighbors. Figure 21a shows a large variance for the average contact time with 20 devices, while the variance gets smaller with increasing number of devices, suggesting that the connection between devices in a denser scenario is more stable. Figure 21b shows that the number of the neighbors increases with an increasing number of devices in the network. All together, these two observations confirm that for our evaluation, a simulated scenario with 20 devices represents a sparse network, while a scenario with 80 devices represents a dense one. With the simulated scenarios as described, we implement our proposed handover strategies and assess their performance under varying configurations of the simulation parameters. Table 8 summarizes the most important parameters for the evaluation.

According to the QoS requirements defined in Section 4.1, we use five metrics for the evaluation defined as follows:

1. *Success rate*: in our simulation, after completing a computing tasks, the operator devices will deliver the result back to the delegator (for statistics collection). Hence, the *success rate* is defined as the ratio between the number computing tasks that can be successfully completed as well as delivered to the delegator against the total number of tasks generated in the beginning of the simulation.

2. *Communication overhead*: we use the total number of ATMT messages which are generated during the simulation run as the metric for communication overhead.

3. *Completion time*: we use the elapsed time after the computing tasks are generated by the delegator until all these tasks are completed and delivered back to the delegator.

4. *Jain's index*: similar to the fairness evaluation of crowd sensing tasks distribution, we use the Jain's index calculated as $JI(x_1, x_2, ..., x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n * \sum_{i=1}^{n} x_i^2}$, in which x_i indicates the number of operations executed by node i, as the metric to quantify the quality of load balancing among the handover mechanisms.

5. *Redundancy factor*: we use redundancy factor as the metric for computation overhead. The *redundancy factor* is determined as the ratio between the total number of operations, which are executed redundantly and the initial numbers of operations first generated in the network.

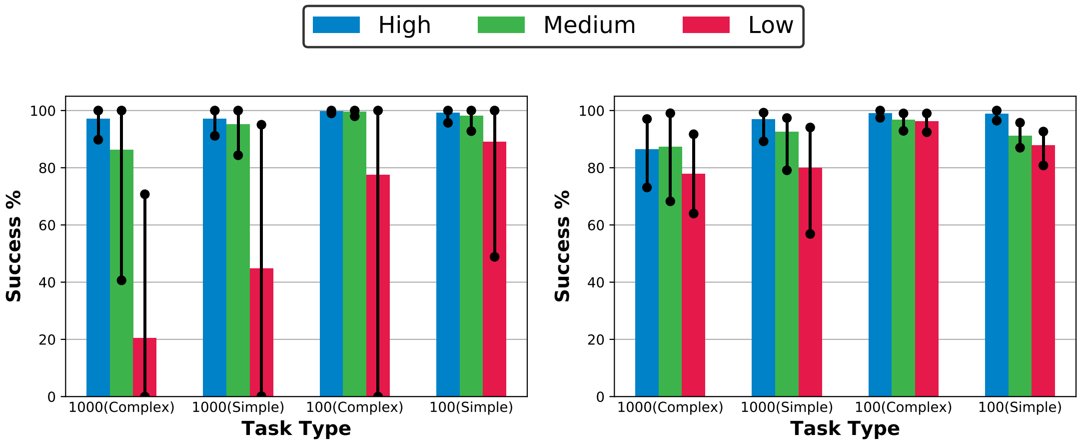
We present the evaluation results for all handover mechanisms and the comparison of their performances in the next sections. All evaluation results are obtained by repeating each configuration ten times. All plots are reported with 95% confidence intervals.

4.6.2 *Analysis of individual Handover Mechanisms*

In this section, we evaluate and analyze each handover mechanism individually to study the performance as well as trade-off of these. Based on the individual analyses, we determine the optimum configuration of each handover mechanism, which is used for the overall comparison later in the next section.

Naive Handover:

The naive handover mechanism is based on epidemic flooding in opportunistic networks. As a result, this mechanism can serve well as the baseline for benchmarking other mechanisms. Similar to the characteristic of epidemic flooding, naive handover obviously generates much overhead due to duplicates of ATMT tasks in the network. We thus leave out the measurement of overhead for later overall comparison. Instead,



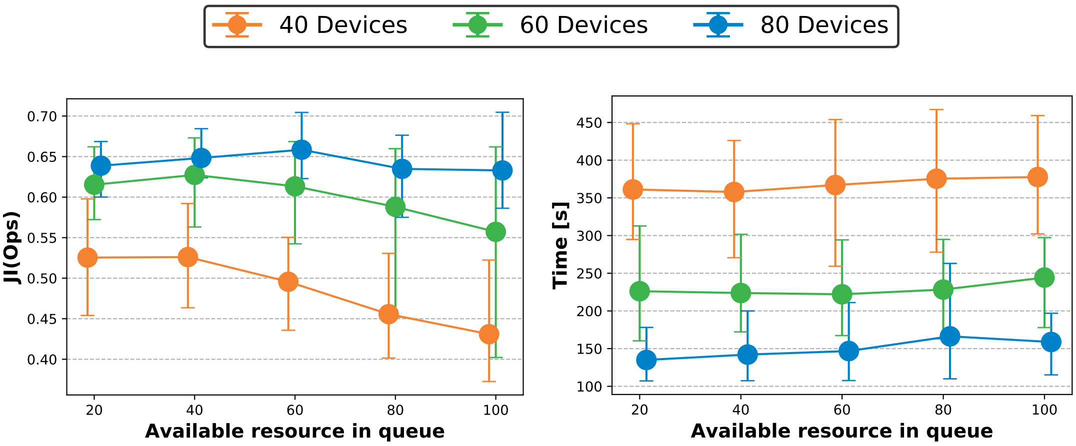
(a) Success rate with 20 devices in the network

(b) Success rate with 80 devices in the network

Figure 22: Success rate of naive handover mechanisms for different types of devices.

we focus on two assessment goals for naive handover. (i) We want to find out the impact of network density on the success rate of complex computing tasks. (ii) Since a naive handover can be regulated by limiting the size of the tasks queue, we want to analyze the effect of limited tasks queues with regards to load balancing and completion time.

To simulate the heterogeneous capabilities of the participating devices, on each operator we generate a number of predefined services that provide the capabilities to execute corresponding operations. For the sake of the evaluation, we create three classes to represent the number of services available on the operators, i.e., *high, medium, low*. There are 5 different services in total, i.e., an operator can execute 5 different operations at best. Services available on the simulated devices of each class are normally distributed. Accordingly, 50% of operators belonging to the *high* class possess 5 services required for executing operations defined in ATMT tasks, while in *medium* class, 50% of operators possess between 2 and 3 services, and in *low* class, 50% of operators possess no service, while the other 50% possess only one single service. Figure 22 presents the results of success rate for a sparse network of 20 devices and a dense network of 80 devices. In both cases, we vary between high and low number of tasks, and between complex and simple tasks. It can be observed that, in case the operators are equipped with a high number of capabilities, the flooding based naive handover can achieve high success rate regardless of network density. However, with low capabilities, the naive handover cannot cope well with a high number of tasks, especially with complex tasks. In a sparse network with 20 devices, operators with low capabilities can achieve a success rate of 20%, while the success rate is on average 77% in a dense network, less than the success rate achieved by operators with more capabilities. This can be explained by the fact that a complex task requires the execution of several operations in a predefined order. Therefore, with less capabilities, the operator possessing the service required for the next operation might not be found in a timely manner. However, in general for naive handover mechanism, the low services availability can be compensated by the high number of operators.



(a) Jain's index of Naive handover

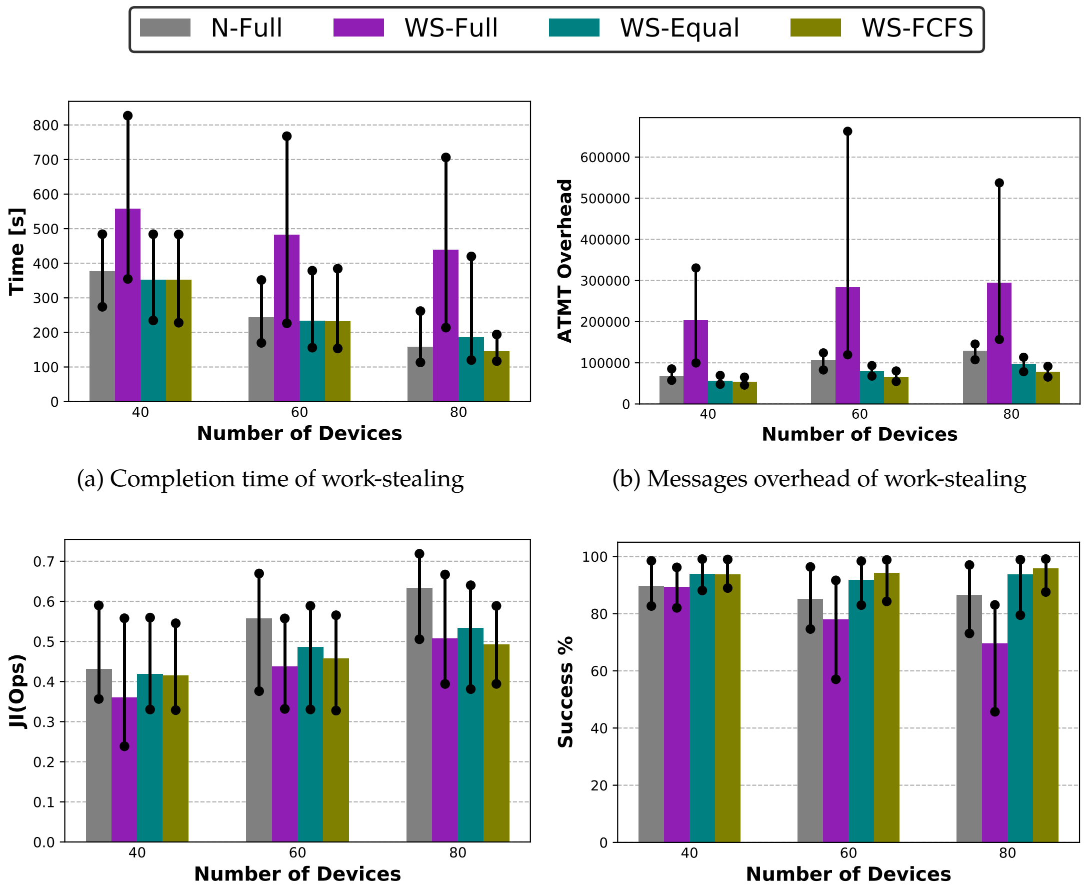
(b) Completion time of Naive handover

Figure 23: Analysis of naive handover mechanisms with regards to available resource in task queue.

We vary the number of acceptable tasks, i.e., the available resources in a tasks queue on each operator to study the effect of limited capacity on naive handover. For this purpose, we determine Jain's index value and measure the completion time with varying network density. The results are shown in Figure 23. It can be observed that the size of the tasks queue does not have any influence on the completion time. A faster completion can be achieved with a higher number of operators. In Figure 23b, the completion time of 80 operators lies on average at 150 s, while 60 operators need around 240 s, and with only 40 operator devices, the network requires more than 350 s to complete all tasks. With regards to Jain's fairness index, Figure 23a shows, that the fairness index values decrease when the available resources in the tasks queue increase. The reason is the greedy execution of operations. Altogether, the results obtained from the analysis of the capacity of the task queue imply, that QoS requirements for distributed processing can be achieved without exhausting the contributed resources of participating devices. Thus, it is possible to reduce the size of the tasks queue without affecting the overall performance, consequently preserving resources of participating devices.

Work-Stealing:

We implement three strategies proposed for the work-stealing mechanism as previously discussed. These strategies are denoted as *WS-Full*, *WS-Equal*, and *WS-FCFS*. The *WS-Full* strategy represents a greedy behavior of participating devices, i.e., when they receive a work-stealing notification, these devices try to offload the maximum number of tasks onto the work-stealing devices. *WS-Equal* strategy lets the work-stealing devices coordinate their neighbors. Thereby, each work-stealing device divides the maximum number of acceptable tasks equally to the number of its neighbors. Hence, the numbers of tasks being handed over locally are regulated by the work-stealing device. The *WS-FCFS* strategy is based on the first come first serve principle, in which the operator that replies first to the work-stealing device, is allowed to handover first.



(a) Completion time of work-stealing

(b) Messages overhead of work-stealing

(c) Jain's index of work-stealing

(d) Success rate of work-stealing

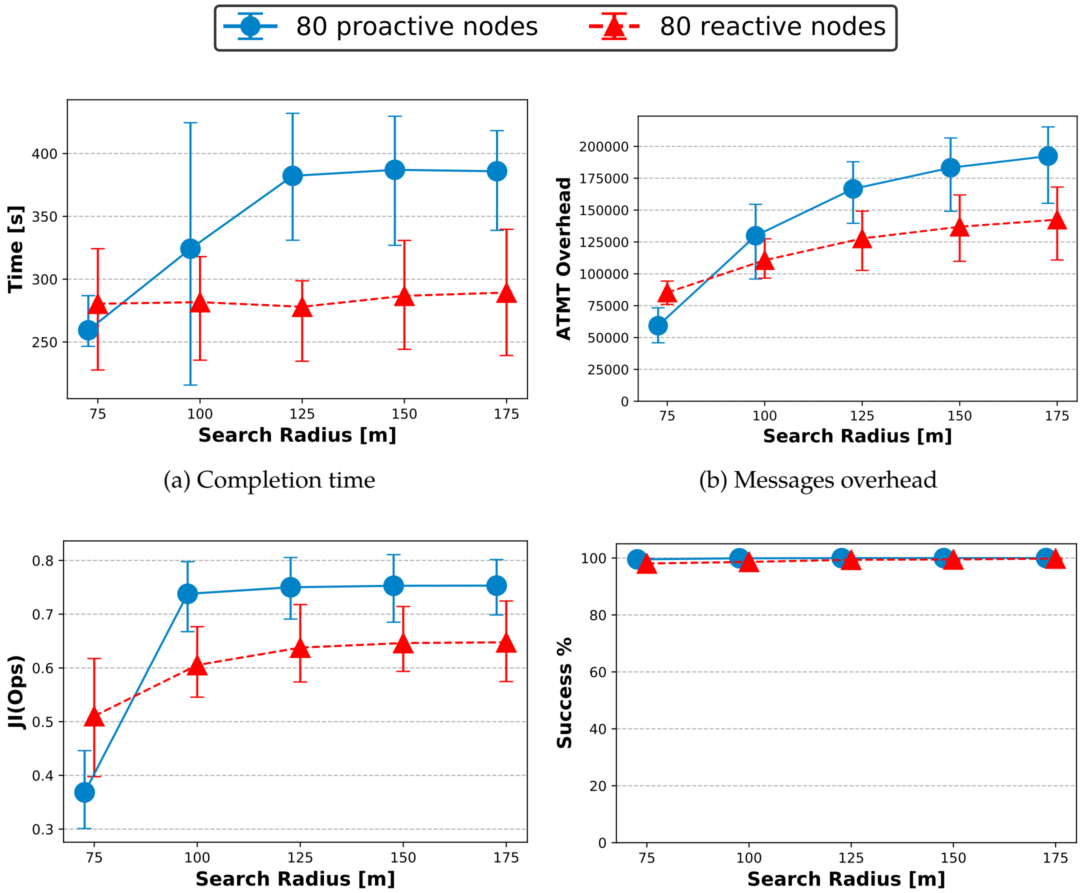
Figure 24: Analysis of work-stealing handover mechanisms.

Contrary to the greedy behavior of *WS-Full*, in *WS-FCFS* each device only offloads the difference of the tasks resided in queue, that exceeds the over-utilized threshold. We compare these three work-stealing strategies against the naive handover with full capacity (denoted as *N-Full*) as the baseline.

Figure 24 summarizes the evaluation results for the work-stealing mechanism. It is obvious, that *WS-Full* with its greedy behavior performs worst compared to all other strategies in all performance categories. This is due to the fact that, *WS-Full* over-utilizes the resource available on each work-stealing devices, therefore, instead of alleviating, *WS-Full* even generates more workload. With regards to completion time, the other two work-stealing strategies *WS-Equal* and *WS-FCFS* perform slightly better. Work-stealing strategies are also able to reduce the communication overhead in terms of the total number of exchanged *ATMT* messages. Interestingly, both work-stealing strategies *WS-Equal* and *WS-FCFS* yield worse Jain's fairness index values compared to the naive mechanism, while being able to improve the success rate especially for dense networks. With 80 operators in the network, the *WS-FCFS* strategy can achieve up to 97% success rate, while the naive handover mechanism can only achieve around 85%. The reason is work-stealing mechanism, in general, is designed to alleviate overburden operators locally, consequently leading to improved success rate as more operators now have idle capacity left to execute operations from ATMT tasks. However, with the effective range of work-stealing mechanism only restricts within one hop neighbors and the chances of work-stealing devices to meet overloaded devices vary as well as highly depend on the network density, the improvement on computation balancing of the whole network cannot be always guaranteed.

Local Optimization:
Last, we evaluate and analyze the handover based on local optimization with two corresponding strategies as previously proposed. The *proactive strategy* triggers a handover process every time an operator device receives new shared context information, which implies an updated list of discovered candidates and potentially new better candidates for task handover. The *reactive strategy* triggers a handover process when a predefined condition is met. We use an overload situation as the triggering condition for the *reactive strategy*. Due to the fact, that the handover based on local optimization searches for candidates as the handover destinations within a search radius, the size of the search radius can effect the overall performance. Accordingly, we measure the evaluation metrics with varying search radius. The evaluation is conducted in a dense network with 80 operators.

The evaluation results for local optimization are summarized in Figure 25. Obviously, increasing the search radius means that the handover of an ATMT task has to traverse through multiple hops before reaching the destination. This intuition can be confirmed in Figure 25b. The total number of ATMT messages increases linearly when increasing the search radius in both proactive and reactive optimization strategies. In the most cases, the proactive strategy generates more overhead than the reactive strategy, due to the fact, that the handover in proactive strategy can be triggered whenever an operator node receives updated shared context information. In a dense network, this can happen quite often. On the contrary, the reactive strategy only triggers in overload situations. Regardless of the search radius, both reactive and proactive strategies always reach 100% success rate, which is explained by the fact that all participating devices always look for the best capable operators for executing the next operation. The completion time for the proactive strategy is longer, due to the fact that the proactive strategy reacts to changes to find better handover destinations, which might occur too often in rapidly changing environments. On the contrary, the completion time of the reactive strategy remains fast and stable, since the reactive strategy is triggered less often. Furthermore, the completion time of the proactive strategy increases with larger search radius, but seems to converge after 125 m search radius. The increasing trend along the search radius is explained by the fact, that it takes longer to reach a handover destination that is located farther away. The convergence trend is explained by the fact, that in a dense network it is highly possible to find the best handover destination within the proximity. The result for the trade-off of longer completion time and of more communication overhead is the improvement in computation balancing in the whole network as visible in Figure 25c. With larger search radius, the fairness index value of the proactive strategy increases as well. However, similar to the completion time,



(a) Completion time

(b) Messages overhead

(c) Jain's index value

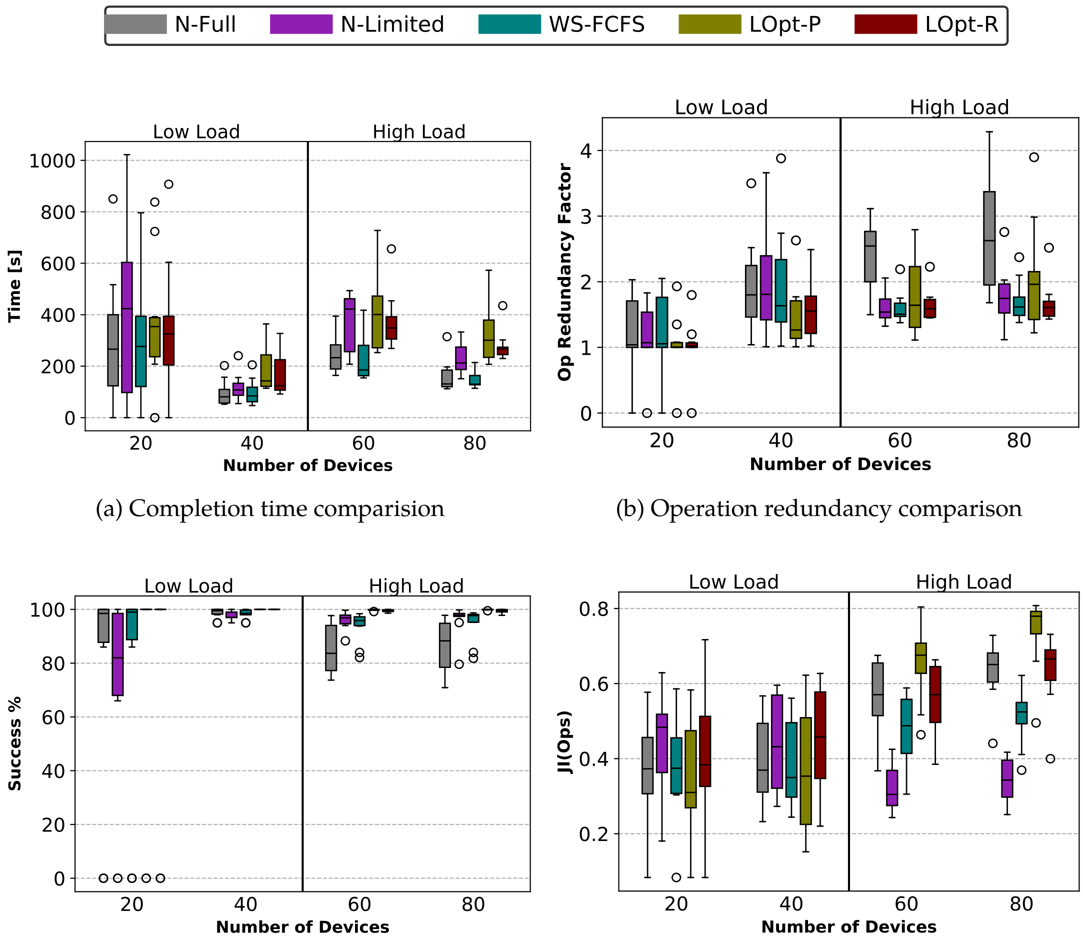
(d) Success rate

Figure 25: Analysis of local optimization based handover mechanisms.

increasing the search radius more than 125 m does not indicate any more improvement with regards to load balancing. Overall, the evaluation results and the analysis of two strategies for local optimization underpins the strength of such mechanism on completing complex tasks. Moreover, it can be confirmed that there is always a trade-off to balance among different QoS requirements.

4.6.3 *Comparison of Handover Mechanisms*

In the previous section, we have analyzed each handover mechanism individually. Our evaluation goal of this section is to provide an overview comparison among handover mechanisms. We use the two mechanisms *N-Full* and *N-Limited* introduced for the naive handover as the baselines for our comparison. Among the three work-stealing strategies, we choose *WS-FCFS* for the overall comparison, since the individual analysis of work-stealing mechanisms has revealed that *WS-FCFS* provides the best performance compared to other work-stealing strategies. Both local optimization strategies are able to provide one of the QoS requirements regarding distributed processing,



(a) Completion time comparision

(b) Operation redundancy comparison

(c) Success rate comparison

(d) Jain's index comparison

Figure 26: Comparison of all proposed handover mechanisms.

therefore, we consider both for the final comparison. In the results shown in Figure 26, the proactive and reactive strategy are denoted as *LOpt-P* and *LOpt-R* respectively.

We conduct the overall comparison among handover mechanisms in both sparse and dense networks. Furthermore, in a sparse network with 20 or 40 operator devices, we only use a low workload in which 100 ATMT tasks are generated and injected. On the contrary, to push for the limit in dense networks, e.g., networks with 60 or 80 devices, we use a high workload in which 1000 ATMT tasks are generated and injected. For the final comparison among mechanisms, we use four evaluation metrics, i.e., completion time, operations redundancy factor to measure the overhead and the efficiency of the handover, success rate, and Jain's fairness index to measure the quality of load balancing. The evaluation results with regards to these metrics are presented in Figure 26.

The completion time observed in sparse networks of 20 nodes is quite high and fluctuates a lot for all handover mechanisms. With more devices in the network, the completion time can be reduced. It can be observed from the results shown in Figure 26a that the completion time of the flooding-based naive mechanism and the work-stealing mechanism are faster than mechanisms based on local optimization. This result is expected, since both flooding based and work-stealing are greedy mechanisms, that trade-off time factor for other *QoS* requirements. This explanation is confirmed when considering the other evaluation metrics. The operation redundancy factor presented in Figure 26b shows that the flooding based naive mechanism with full capacity expectedly executes a single operation multiple times. Other mechanisms, i.e., N-Limited, WS-FCFS, and both local optimization strategies are able to decrease the operation redundancy factor, implying, that the computation resources can be alleviated. With regards to success rate, the results presented in Figure 26c indicate that while the success rate of naive and work-stealing mechanisms fluctuates, both proactive and reactive strategies of local optimization ensure the highest success rate of 100%. The marginal variance in the box plot of the success rate by *LOpt-P* proactive strategy and *LOpt-R* reactive strategy suggests that both strategies are stable and thus always deliver almost 100% success rate. Lastly, the evaluation results obtained for Jain's fairness index are presented in Figure 26d. Evidently, load balancing is difficult to achieve in sparse opportunistic networks. The fairness index values vary a lot with 20 and 40 devices regardless of which handover mechanisms are employed. On the contrary, a dense opportunistic network provides an environment which allows for achieving better load balancing. We can observe, that both strategies based on local optimization are able to deliver a high fairness index value in a dense network. The proactive strategy achieves 0.65, while the reactive strategy only achieves around 0.58 with 60 operators. The highest fairness index value can be achieved by the proactive strategy of the local optimization with 80 operators. In this case, the proactive strategy is able to achieve a fairness index at around 0.8.

All in all, the evaluation results show that it is possible to satisfy QoS requirements, using only locally shared context information and distributed coordination.

4.7 DISCUSSION

In this chapter, we proposed a concept for distributed processing, specially designed for mobile opportunistic networks. To this end, we designed the adaptive task-oriented message template, termed ATMT, as the enabling technique for distributed processing of complex computing tasks, which require multiple processing stages in a decentralized environment. Based on the construction of the ATMT, we proposed several handover mechanisms, which let a device offload a processing task to the others, aiming to achieve QoS requirements as specified by the task. Our proposed handover mechanisms do not rely on any centralized coordination and only leverage the context information shared by participating devices locally. Overall, using the tasks message template ATMT with the handover mechanisms allows the participating devices to make autonomous decisions. In this manner, a decentralized cooperation and distributed coordination among devices to accomplish a processing task is enabled. The mechanisms discussed in this chapter provide a basis to create distributed processing applications, which can be utilized in a disaster relief situation to provide emergency response services even when the communication infrastructure is impaired. For instance, *Person Finder* service provided on Google platform to let a user inquire about the status of a person, can be realized by face detection technique through distributed image processing based on ATMT message template as shown in our demonstration [133, 135]. In general, this sample application demonstrates that information can be extracted from crowd collected data through distributed processing, leveraging idle resources available on participating devices. With regards to our research goals, the distributed processing model presented in this chapter constitutes the second objective of the information retrieval workflow. With the mechanism to distribute crowd sensing task and the distributed processing mechanism to process crowd collected data, the results need to be delivered to the *information consumers* reliably, despite the rapid changes and the uncertainty in mobile opportunistic network. In the next chapter, we present our mechanism to deliver results to conclude the information retrieval workflow.

5 RESULTS DELIVERY FOR MOBILE INFORMATION CONSUMERS

Results delivery is the last step of information retrieval. In the first step, crowd sensing tasks are distributed to appropriate information producers where they trigger the data collection. Depending on the request of information consumers, data can be sent back either in raw measurements or being processed during the forwarding phase. Consequently, we refer to both processed data and raw measurements as *results* of sensing tasks. In an emergency response scenario, requested data can either be relevant for the requestor only or it can be relevant for a larger group of users, e.g., within a given area. Whether the results of the sensing tasks should be disseminated to many information consumers or should be delivered to individual information consumers, depends on the requirements of the crowd sensing applications and requests. Despite the fact that NDN-based networks provide a built-in feature for data dissemination through in-network caching [4, 203], the support for results delivery to individual information consumers is limited. Since results delivery is a subclass of data dissemination, NDN-based forwarding approaches achieve results delivery in the same way as data dissemination.Thereby, NDN-based approaches cache requested data on devices participating in the data forwarding phase. The results are delivered to a mobile information consumer by having this consumer send out the same sensing task. If the results are only relevant for a single information consumer, then caching data on many devices becomes redundant for in-network storage and consumes resources, which are scarce considering NDN-based mobile opportunistic networks and disaster situations.

In this chapter, we tackle the problem of *results delivery* for mobile consumers as the final step to complete *information retrieval*. In Section 5.1, we discuss the challenges and requirements of results delivery for mobile consumers. We introduce our mechanism to deliver results in Section 5.2. Thereby, we present the constructions of both interest and data packets to support results delivery. We rely on mobility prediction of mobile consumers for results delivery. In Section 5.3, we elaborate on our approach to integrate distributed data processing into the data forwarding phase. Finally, we present the evaluation in Section 5.4, which consolidates all three aspects of information retrieval.

5.1 REQUIREMENTS AND CHALLENGES

We consider two aspects: a data forwarding mechanism to deliver results as being requested by information consumers and the integration of distributed data processing during the forwarding. The requirements and challenges for decentralized crowd sensing tasks distribution in Section 3.1 and for distributed data processing in Section 4.1 also apply for results delivery. The consolidated requirements are:

Quality requirements: The objective of results delivery, in contrast to data dissemination [19], is to forward the results to particular information consumers. In Chapter 3, we introduced our mechanism to distribute crowd sensing tasks. Thereby, we ensure that the sensing tasks reach capable information producers, which provide the data that satisfies the quality requirements for the tasks. Thereupon, the results of crowd collected data need to be delivered to the consumers in a timely manner. In emergency response scenarios, fast delivery of collected data is particularly important, since such data are used by first responders to react to the current situation.

Cost for results delivery: Careful utilization of resources in terms of residual energy is important to maintain communication networks formed by mobile devices and to prolong services which support relief work of the first responders in disaster situations [94]. To enable data dissemination in information-centric networks in general, and in NDN-based networks in particular, in-network caching is leveraged which allows a mobile consumer to retrieve results from caches on participating devices. As discussed in Chapter 3, communication on NDN-based opportunistic networks relies on broadcast. As a result, many devices can obtain a copy of the data and store the data in their local storage by default. On the one hand this characteristic is beneficial for dissemination if the data is relevant for multiple consumers. On the other hand for results delivery to individual information consumers, redundant in-network storage introduces significant overhead. A directed data forwarding for *results delivery* is, thus, more beneficial to reduce resources consumption and communication overhead compared to broadcast based approach for dissemination [191].

Incentive for participants: In Section 3.1, we elaborate on the incentive for participants to contribute their resources to forwarding. Providing services and applications which benefit community or a group of users can be considered as a form of incentive [218]. In an emergency response scenario, both data dissemination and results delivery are beneficial for many people. While data dissemination can be used for notifications of the authorities, results delivery ensures the delivery of sensing measurements for the authorities, so that they can assess the current situations and plan their relief operations more efficiently, which in turn benefits all people. Thus, with regards to emergency situations we assume that the participants are fully collaborative and willing to contribute their resources for both results delivery as well as distributed processing.

The following challenges for results delivery are inherent in mobile opportunistic networks:

Decentralized Communication: The communication in mobile opportunistic network is decentralized. Consequently, results delivery cannot rely on central coordination entity to track the movement of mobile information consumers. Thereby, a distributed scheme to track and share the mobility of mobile information consumers for targeted result delivery is required.

Heterogeneity: Results delivery is a combination of data forwarding and distributed data processing during the forwarding. Consequently, we have to consider the heterogeneity of participating devices with regards to their resources, residual energy, and their capabilities. Data forwarding consumes resources and energy of participating devices regarding communication overhead and in-network storage. Thus, forwarding devices have to be chosen considering their heterogeneous resources carefully. Additionally, the goal of the distributed data processing while forwarding data might require multiple and special operations to complete. Therefore, the utilization of heterogeneous capabilities available on forwarding devices to accomplish a complex processing task is desired.

Uncertainty: The mobility of devices in an opportunistic network leads to uncertainty when designing a forwarding approach for results delivery, since the status of mobile consumers can change over time. Therefore, a distributed scheme to deliver results to mobile consumer needs to be adaptive to the changes of the respective consumers. Accordingly, participating devices need to be able to make autonomous forwarding decisions to cope with the changes of mobile consumers as well as the rapid changes in their surrounding environment.

5.2 DATA DELIVERY FOR MOBILE CONSUMERS

In the overall workflow of our NDN based information retrieval elaborated in Section 3.3, we considered static information consumers. In this chapter, we consider results delivery for *mobile* consumers, i.e., a consumer moves away from the starting position after sending out crowd sensing tasks. With the current data forwarding mechanism as discussed in Section 3.3, a mobile consumer can propagate new crowd sensing tasks, requesting for the same data. Thereby, the new crowd sensing tasks do not necessarily have to be forwarded to the information producers again, since data can be served by devices that already cached the results. However, the default data forwarding mechanism relies on flooding; as a result, data forwarding leaves many copies of the data in an NDN-based opportunistic network. This does not scale and generates significant overhead since the requested data is only relevant for individual mobile information consumers. In [191, 192], the authors consider a similar problem for vehicular networks and rely on *breadcrumb* messages to allow the response to track and to follow the mobile information consumers. However, their *breadcrumb* based routing is not designed for NDN-based networks. Furthermore, *breadcrumb* based routing still generates much overhead for keeping *breadcrumb* messages available. State of the art for data forwarding in NDN-based V2V network [4] only focuses on pushing data packets to mobile consumers as fast as possible. The communication overhead in terms of data packets is reduced by restricting the number of allowed hops. However, this approach still relies on flooding and therefore generates much redundant overhead for in-network caching.

Focusing on results delivery for individual mobile information consumers, we utilize mobility prediction to estimate future locations of mobile information consumers in order to guide data forwarding directly towards the respective consumers. Based on predicted locations, the information producers can restrict the data forwarding towards the direction of mobile information consumers. With such directed data forwarding, we aim to reduce communication overhead while achieving timely delivery for mobile consumers. Similar to the interest forwarding phase, we also rely on distributed coordination to make forwarding decisions during the data forwarding phase. Thereby, we again embed context information as attribute fields in both interest and data packets, which let participating devices share their status and make autonomous decisions benefitting results delivery.

5.2.1 *Interest and Data Packets Construction*

To enable results delivery for data forwarding, we need to embed context information in both interest and data packets. The reason is, mobile consumers need to share their context information through *interest packets* to allow for mobility prediction and forwarding devices need to use context information in *data packet* to make autonomous forwarding decision to deliver results. The constructions of interest and data packets required for results delivery are illustrated in Figure 27.

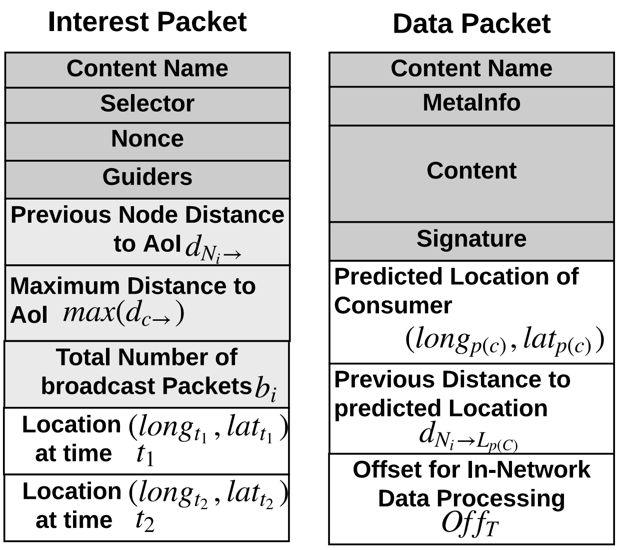


Figure 27: Extended interest and data packets with attribute fields required for results delivery based on mobility prediction and for distributed data processing. The white fields are additional attribute tags used in this chapter.

The modified interest and data packets contain default fields as defined for NDN-based network, attribute fields required for interest forwarding in crowd sensing tasks distribution as proposed in Chapter 3, and further additional fields required for both results delivery and distributed data processing (white fields in Figure 27). We embed past locations of mobile consumers into interest packets, which are used as inputs to predict future location of mobile consumers for data forwarding. To allow for mobility prediction, at least two past locations with time stamps are required. In general, the prediction for future location might be more accurate with more past locations. For instance, in [40], the authors use a Markov predictor with 100 past location records to predict future connections of nodes for routing in a MANET. However, exhausting interest packets with too many additional attribute fields obviously generates much overhead on opportunistic networks. Consequently, we decide to trade-off the accuracy for mobility prediction for less communication overhead. We use two additional fields to contain two past locations of mobile information consumers. Each interest packet is extended with two additional attribute fields containing two Cartesian coordinates transformed by Mercator formula [113] from the longitude, latitude coordinates of mobile consumers at two time points t_1 and t_2. The uncertainty and possible dynamic changes over time will be handled by the forwarding strategy, which we discuss in the following section.

In data packets, we include three attribute fields to facilitate the data forwarding phase for results delivery as well as to allow for distributed data processing. These attributes are: (i) predicted location of *mobile information consumers*, which is represented through the predicted coordinates, (ii) the distance from the previous data forwarding device towards the predicted location of consumers, and (iii) an *offset* $\mathtt{Off}_T$ required to track *processing state* which is required for computation handover of complex processing tasks as proposed in Chapter 4.

5.2.2 *Data Forwarding based on Mobility Prediction*

In this section, we elaborate on (i) how mobility prediction can be integrated in NDN-based opportunistic networks and (ii) how to use mobility prediction to guide the data forwarding phase for results delivery.

According to our model to distribute crowd sensing tasks to retrieve information (cf. Chapter 3), mobile information consumers that want to retrieve data from an AoI, propagate interest packets in an NDN-based opportunistic network to search for information producers. For results delivery, in addition to the context information required for two-phase interest forwarding mobile information consumers also need to share their past locations with the information producers. Accordingly, after obtaining the requested data information producers can use the embedded past locations of mobile consumers to predict their future locations and to trigger the data forwarding phase to deliver the results. Thus, before propagating interest packets, mobile consumers determine their two most recent locations. We assume that devices in our system are able to determine their locations e.g. via GPS, and these devices can store past locations locally for later use. We utilize only the two most recent locations of mobile consumers in order not to generate much overhead through embedded shared context information as previously discussed. As a result, the interest packets propagated by mobile consumers contain additional attribute fields as shown in Figure 27.

The parameters used for results delivery based on mobility prediction as well as for distributed data processing during forwarding are summarized in Table 9. To avoid collision of interest and data packets defer timers are again required in the data forwarding phase. Here, the parameters required for determining defer timer were introduced in Table 2 in Section 3.3. These parameters are used again for calculating the defer timer for data forwarding. However, instead of considering the coordinates of the AoI as the destination, in the data forwarding phase the coordinates of the predicted locations for mobile consumers are used.

Table 9: Parameters for the data forwarding based on mobility prediction and distributed data processing.

Parameters	**Meaning**
$(loc_{x_{t_i}}, loc_{y_{t_i}})$	the Cartesian coordinates of a mobile consumer at time t_i
$(loc_{x_{t_p}}, loc_{y_{t_p}})$	the Cartesian coordinates of predicted location for a mobile consumer C at time t_p, used as the destination for data forwarding
v_C	average velocity of a mobile consumer C
$d_{N_i \to L_{p(C)}}$	distance between the device N_i and the predicted location of a mobile consumer C
Off_T	offset used to track the processing state of distributed data processing

When information producers receive an interest packet, they can extract the two past locations of the respective mobile consumer $(loc_{x_{t_1}}, loc_{y_{t_1}})$ and $(loc_{x_{t_2}}, loc_{y_{t_2}})$, sampled at two time points t_1 and t_2 (with $t_1 < t_2$) respectively. Based on these two past locations and their corresponding timestamps, the future location represented by coordinates $(loc_{x_{t_p}}, loc_{y_{t_p}})$ of mobile consumer C at time t_p is estimated using the formula proposed by Shah et al. [174] as follows:

$$loc_{x_{t_p}} = loc_{x_{t_2}} + \frac{v_c(t_p - t_2)(loc_{x_{t_2}} - loc_{x_{t_1}})}{\sqrt{(loc_{x_{t_2}} - loc_{x_{t_1}})^2 + (loc_{y_{t_2}} - loc_{y_{t_1}})^2}} \tag{13}$$

$$loc_{y_{t_p}} = loc_{y_{t_1}} + \frac{(loc_{x_{t_p}} - loc_{x_{t_1}})(loc_{y_{t_2}} - loc_{y_{t_1}})}{loc_{x_{t_2}} - loc_{x_{t_1}}} \tag{14}$$

In the above equation, v_C represents the average velocity of mobile consumer. To obtain v_C at information producers, two options are possible. (i) The velocity can be determined by a mobile consumer itself and is embedded as another attribute field in the interest packets. (ii) The velocity can be estimated using the two past locations shared in an interest packet. Thereby, the distance between two locations are considered as the travel distance, and the elapsed time between two timestamps is the travel time. As such, v_C is estimated as follows:

$$v_C = \frac{\sqrt{(loc_{x_{t_2}} - loc_{x_{t_1}})^2 + (loc_{y_{t_2}} - loc_{y_{t_1}})^2}}{t_2 - t_1} \tag{15}$$

The embedded attribute fields received by the information producers also reveal whether the requested data are meant for dissemination or for results delivery. In our system, if information producers cannot find any embedded past locations of information consumers in interest packets, then the requested data is disseminated to all devices. This can be utilized for example by the authorities in emergency situations to send a request assessing the danger level of an AoI. The results of this request are disseminated directly during the data forwarding phase to as many people as possible, to notify them to avoid the danger present in the AoI. Thereby, data packets are propagated to the information consumers without any restriction. On the contrary, if past locations of an information consumers are found in an interest packet, information producers determine that, the the respective information consumer is mobile and the data should be delivered only to this consumer.

In case the data forwarding occurs in the *results delivery* phase, information producers extract the two past locations from an interest packet, and predict the mobility of the respective mobile consumer according to Equation 13, and 14. As devices, including the mobile consumers, are highly mobile, we need to handle the uncertainty for mobility prediction during the data forwarding phase. Thereby, the first cause for uncertainty is the deviation from the predicted location when choosing the time parameter t_p. In an opportunistic network, it is impossible to estimate the exact time t_p when data packets can reach the mobile consumers. The second cause for uncertainty are changes in movement of the respective mobile consumers, e.g., velocity change, or moving direction change. Accordingly, to handle the uncertainty of mobility prediction for consumers, we propose several strategies which can be combined together as follows: (i) naive estimation of t_p, (ii) *breadcrumb* based correction through reissuing interest packets with updated shared context information, and (iii) utilization of the *buffer-zone* concept proposed for our two-phase interest forwarding as a tolerant threshold for the predicted destination. The first and second strategies which have the goal to counter uncertainty are based on [174]; the concept in [174] however is designed focusing on MANETs with established routes.

For the first strategy, we assume that mobile consumers always sample the latest location before propagating interest packets. Accordingly, the timestamp t_2 extracted from an interest packet can be used as the begin of interests transmission from mobile consumers. We determine the elapsed time since a mobile consumer sends out interest packets until these packets are received by information producers at time $t_{current}$ as $t_{current} - t_2$. Despite the fact, that the transmission time of the interest forwarding phase does not equal to the transmission time of the data forwarding phase in an opportunistic network, the calculated time can still be used as a naive estimation. Consequently, t_p is estimated as: $t_p = t_{current} + (t_{current} - t_2) = 2 \times t_{current} - t_2$. Predicting the future location of a mobile consumer and forwarding the data towards the direction of the predicted location reduce the communication overhead compared to general *breadcrumb* based routing such as [191] or to default data forwarding such as [4].

In the second strategy, we leverage the built-in *breadcrumb* feature of NDN-based opportunistic networks, enabled by the *Content Storage* of participating devices. By monitoring its own location, a mobile consumer defines a threshold, when its current location deviates too much from the predicted location (due to changes in velocity or moving direction). If the deviation exceeds a predefined threshold, the respective mobile consumer propagates interest packets containing the same name of previous sensing tasks and embeds new past locations into the interest packet to update its new moving pattern. When the new interest packet arrives at a forwarder device that already caches the results, the respective forwarding device uses the name of the interest packet and an embedded ID value to determine that the respective mobile consumer has changed its moving direction. As a result, this forwarding device extracts the updated past locations from the newly received interest packet and predicts the new location of mobile consumer to adjust the data forwarding accordingly.

For the third and last strategy, we use the buffer-zone concept to define a region around the predicted location for mobile consumers as a geo-destination. The size of the buffer-zone can vary based on the density of the network, which can be estimated by overhearing broadcast transmission of neighbors [211].

With the predicted location of mobile consumer $L_{p(C)}$, we base the *data forwarding* on our proposed two-phase *interest forwarding* concept introduced in Chapter 3. To deliver the results we again rely on the broadcast and rebroadcast primitives of NDN-based opportunistic networks. Thereby, the forwarding devices make autonomous forwarding decisions, aiming to carry the data packets closer to the predicted location of mobile consumers. To this end, we include two attribute fields in the data packets, i.e., (i) the coordinates of the predicted location for a mobile consumer and (ii) the distance of the current forwarding device towards the predicted location. In default NDN networks, if the name of a data packet cannot be found in the Pending Interest Table (PIT), the packet is dropped. However, in NDN-based opportunistic networks, there is no end-to-end path between information consumers and producers. In sparse opportunistic networks, the chance to meet another forwarding device is also very low. Therefore, to increase the chances for results delivery we adapt the default behavior for *content store*. In our system, a forwarding device also caches unsolicited data temporarily in its content store. During the data forwarding phase, upon receiving a data packet a forwarding device extracts the predicted location of the respective mobile consumer and the distance from the previous forwarding device towards the predicted location. A forwarding device decides to rebroadcast, if it determines itself to be an eligible data forwarder, which holds true if all of the following conditions are met. (i) The current distance from the corresponding device towards the predicted location is less than the distance from the previous forwarding device, which indicates the current forwarding device is closer to the predicted location. (ii) The corresponding device is moving towards the direction of the predicted location, which is characterized through an angle threshold similar to Equation 4 in Chapter 3. (iii) The current velocity and residual energy level of the corresponding device exceed a defined threshold. Similar to our interest forwarding approach, we use distance, moving characteristic, and residual energy to make sure that only the best forwarders are chosen to rebroadcast data packets. Our aim for data forwarding based on mobility prediction is to deliver the results fast and to reduce communication overhead. To avoid collision of data packets, we determine a defer timer for data forwarding. The defer timer is calculated in the similar manner as for defer timer used in our two-phase interest forwarding approach as:

$$DT_{Data} = Time_{DeferSlot} * (T_d + T_e + T_s + T_{md}) + T_{Random} \tag{16}$$

in which, T_d, T_e, T_s, T_{md} are defined similarly as in Equation 6. The distance component T_d for results delivery is calculated as $\frac{d_{max} - d_{N_i \to L_{p(C)}}}{d_{max}}$. Thereby, d_{max} is the distance from the information producer towards the prediction location of the targeted mobile consumer and $d_{N_i \to L_{p(C)}}$ is the current distance from the corresponding device N_i towards the predicted location $L_{p(C)}$. With regards to the moving direction, the component T_{md} for results delivery is calculated based on the current movement characteristic of the forwarding devices and buffer-zone threshold set for the predicted location accordingly. Determining the defer timer for data packet in this way ensures, that if a forwarding device is closer to the predicted location of mobile consumers, the corresponding device broadcasts faster, thus reducing the results delivery time and increasing the chance that the data packets can be delivered to mobile consumers at the predicted location.

5.3 INTEGRATION OF DISTRIBUTED DATA PROCESSING

In this section, we present the integration of distributed data processing as introduced in Chapter 4 into results delivery. Accordingly, crowd collected data can be processed during data forwarding. Thereby, information can be extracted directly within opportunistic networks before reaching information consumers. As discussed in Section 4.1, to facilitate distributed data processing in opportunistic networks we rely on distributed coordination and autonomous decision of participating devices. The essence of the *task message template* proposed in Chapter 4 is to bundle meta-information and payload data into a single message, so that each device is able to deduce the current state of the processing tasks and to execute operations on data without centralized coordination. Therefore to realize distributed data processing, the data packets of NDN-based networks are adjusted to encapsulate processing information additionally to payload data.

5.3.1 *Naming Scheme and Data Packets for Processing*

To facilitate the complete *information retrieval workflow* consisting of sensing tasks distribution, distributed data processing, and results delivery, we define the following naming scheme as an extension for the naming scheme proposed in Definition 3.1 as follows:

Definition 5.1: Naming Scheme for Data Processing Intergration

<Crowd-Sensing-Tasks>/operation-x/operation-y/operation-z

In the above definition, *"<Crowd-Sensing-Tasks>"* contains the same naming scheme defined in Definition 3.1 which expresses which type of data are collected, while *"operation-x/operation-y/operation-z"* implies the operations required for a complex processing task. The names of the operations can be chosen from a standard set to ensure interoperability among participating devices. Combining the definition of crowd sensing tasks and the definition of processing tasks into a single naming scheme allows us to specify *information retrieval* as a single task. The *information retrieval* task is accomplished by the participating devices collaboratively.

Since a processing task such as in-network data analysis is in general complex and might require several operations, the hierarchical naming scheme is suitable to describe such complex processing task. Based on the Named Function Networking (NFN) proposed by [193], we use "/" to separate functions/operations in the naming scheme from each other. The combination *"operation-x/operation-y/operation-z"* constitutes a single complex processing task, which is completed by executing operations x, y, z in the specified order given by the naming scheme. As an example, an image processing pipeline to detect victim in a disaster situation can be expressed as "*edge-detection/object-detection/victim-detection*" as shown in [134]. A hierarchical naming scheme for distributed data processing allows us to specify the granularity requirements on the processed results. A longer naming scheme for distributed processing implies more operations, suggesting more fine-granular requirements on the processed results.

To enable computation handover of partially processed tasks during data forwarding, we require a mechanism to share the state of the processing task. In this context, the processing state is defined as how many operations from the processing task have been executed on the corresponding data. In the task-message template in Chapter 4, this can be achieved by the *checksum* field and by traversing the *operations graph* to look for the next unexecuted operation. To embed the processing state into data packets, we introduce an offset $\mathtt{Off}_\mathtt{T}$ attribute field (cf. Figure 27). As suggested by its name, this attribute field holds the offset of the next unexecuted operation from the end of the naming scheme of the processing task. A zero offset implies that the processing task is completed.

Altogether, the proposed naming scheme and the offset attribute tag in data packets allow us to integrate our distributed data processing mechanism based on self-encapsulated task message into the data forwarding phase in NDN-based opportunistic networks.

5.3.2 *Processing Model*

In this section, we elaborate on the distributed processing model based on the naming scheme and data packets during the data forwarding phase.

Figure 28 shows the integration of a processing module into the data forwarding phase in NDN-based opportunistic networks. By default, upon receiving a data packet, a forwarding device checks if the data is matched to any interest stored in the PIT table. A forwarding device drops a data packet, if this data is not associated with any cached

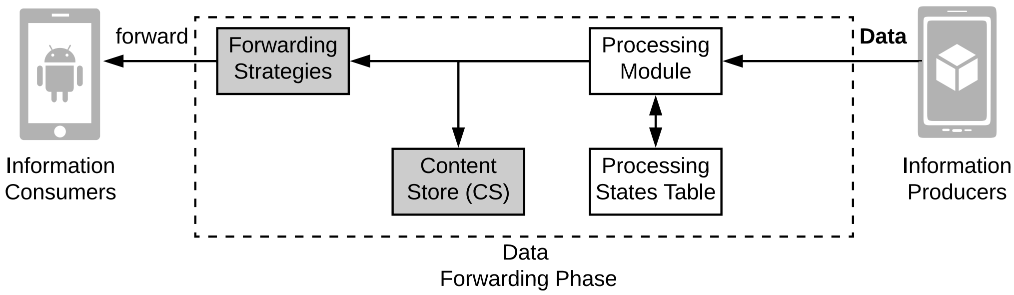


Figure 28: Integration of distributed data processing in NDN-enabled forwarding devices for in-network processing

interest in the PIT table. To integrate distributed data processing into results delivery, a device which wants to contribute to distributed data processing does not always hold a matched interest inside its PIT table. Furthermore, high mobility makes it difficult to find capable devices to execute an operation. As a result, we do not route the data through PIT table on forwarding devices as in default NDN. Instead, we introduce a *processing module*, which receives data packets directly from the communication interface and executes the operations defined in the processing task specified in each data packet if necessary. The execution of operations is based on the processing model as proposed in Chapter 4. The results of data packets obtained by executing the operations are routed to *content store* for caching and also to *strategies module* to forward the processed data packets further.

Based on the name of the operations specified in the data packet, the processing module determines how to proceed with the data packet. Thereby, if a forwarding device wants to contribute to distributed data processing and still has computing capacity, this device searches for the next unexecuted operation and check if can execute the respective operation. This process is repeated until the corresponding device cannot execute the processing task further or when the resources available on this device are at low level. The processed data are stored in the content store before being forwarded. Thereby, the data packets contain the offset $\mathtt{Off}_T$ corresponding to the state of the processing task, so that the next forwarding devices can continue on the respective processing task. To satisfy the quality requirements of distributed data processing during the forwarding, we refer to our processing handover strategies elaborated in Section 4.5.

Since the content store on an NDN-enabled device removes all additional attribute fields before caching the data packets, additionally to the content store we introduce a new data structure—*processing states table*—to store the offset associated with the name of a data packet and its respective processing task. The *processing states table* is required to remember the states of different processing tasks associated with the names of stored data on each device. As an example, an entry in the *processing states table* which contains the name "*op-1/../op-n*" with an offset one indicates that, the data associated with the given name has been processed up to the final operation *op-n*;

consequently, to continue as well as to complete the processing task, *op-n* will need to be executed next.

This data structure also facilitates replacing outdated processed data, dropping or merging data packets with similar naming schemes or similar processing tasks when necessary. Thereby, our goal is to reduce redundant data packets in the network. We merge processing tasks and replace outdated data based on two factors: (i) the naming scheme of the operations sequence defined in each data packet, (ii) the processing state extracted from the operations offset embedded in each data packet. If the specifications for the crowd sensing tasks in the naming scheme of two data packets are not matched, then these are two different requested data types and thus cannot be merged. In case two data packets contain the same naming scheme, we preserve the data packets with smaller processing offset, since such data packets indicate a better processing state near to completion. Similarly, if the results delivery time needs to be low, data packets with shorter name prefix are preserved, since these contain less operations. However, the participating devices can also agree on a common merging strategy to ensure a particular goal, e.g., preserving longer names to favor more fine-granular processing task. Overall, merging data packets can be beneficial towards ensuring and improving the quality requirements specified by the information consumers.

5.4 EVALUATION

In this section, we present a consolidated evaluation, considering all three aspects of information retrieval workflow, namely, crowd sensing tasks distribution, distributed data processing, and results delivery. The evaluation discussed in this chapter extends the evaluation presented in Section 3.4. Back in Section 3.4, our main goal was to assess the performance and the limit of decentralized crowd sensing tasks distribution, especially, in case of chaotic, highly mobile devices. For the consolidated evaluation, we consider a mobility model, which resembles human like mobility patterns, representing mobile devices in emergency response situations.

5.4.1 *Evaluation Setup*

We rely on NDNSim network simulator [1] for the consolidated evaluation. Thereby, we extend our simulation presented in Section 3.4 with the following components: (i) we use *truncated Levy walk* mobility model, which is reported by Lee at al. [160] to resemble human mobility, (ii) we implement our *results delivery* approach based on mobility for mobile consumers as introduced in this chapter, (iii) we implement distributed data processing during the data forwarding phase as previously discussed. Furthermore, we augment the simulation scenario with mobility for information consumers for the sake of evaluation with results delivery. The structure of the extended simulation scenario is illustrated in Figure 29.

Most of the simulation parameters remain the same from the setup of crowd sensing tasks distribution in Section 3.4. We use the simulation area with 800×800 m^2, the AoI with radius size of 50 m, 12 information consumers, 25 information producers, and

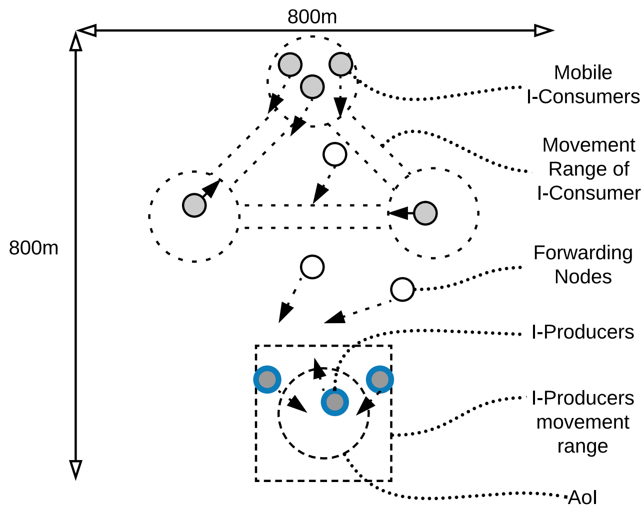


Figure 29: Illustration of the simulation scenario for consolidated evaluation. Information consumers are mobile.

several forwarding devices (ranging from 20 to 100 nodes). We create three *movement spots* to control the movement range of the information consumers. The *movement spots* are positioned as dashed circles within *the movement range of information consumers* shown in Figure 29. At the beginning of the simulation, each mobile consumer starts near the top of the simulation area. Thereafter, each mobile consumer chooses one movement spot randomly but different from its current position as its next target destination, while propagating interest packets as crowd sensing tasks. After reaching a destination, a mobile consumer does not stop and continues its movement in the same manner. The mobile consumers move between two movement spots with a random velocity between 15 and 20 m/s. We choose such high velocity for mobile consumers to challenge and to assess the performance of *results delivery* based on mobility prediction. In this scenario, the mobile consumers represent the authorities such as firefighters with firetrucks in emergency response scenario, that need to move around quickly to coordinate relief operations.

As previously mentioned, we use the *truncated Levy walk* mobility model for the forwarding devices. We rely on BonnMotion [10] to generate 20 traces according to this model in order to simulate the appropriate mobility of pedestrians. For the evaluation of fast forwarding devices, we refer to the results presented in Section 3.4. The traces are installed on the forwarding devices in the simulation and define the characteristics of their movements. In Section 4.6, we analyzed the characteristics of mobility traces generated by Levy walk model. To recapitulate from the measurements shown in Section 4.6 and Figure 21, the forwarding devices with the Levy walk traces are distributed equally on the simulation area. The devices have more neighbors and the contact duration is more stable with increasing number of devices. Furthermore, we measured the movement speed of devices with Levy walk and observe that several

Table 10: Simulation parameters for consolidated evaluation

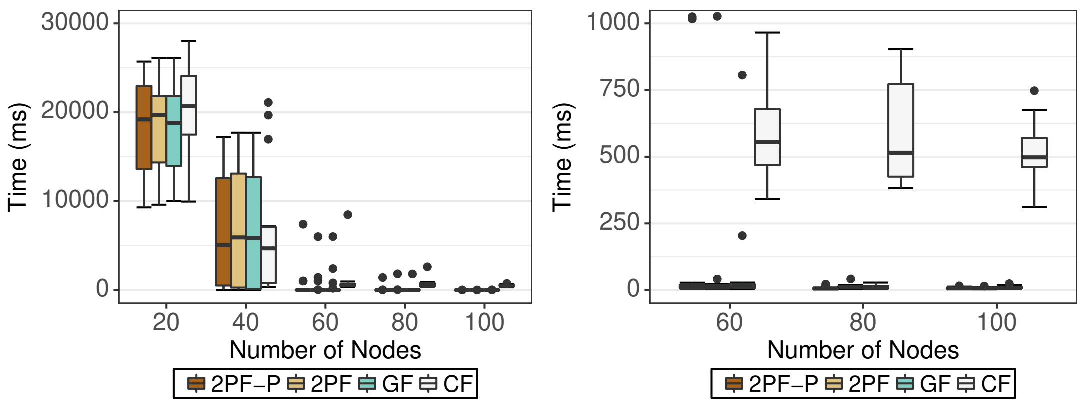
Parameter	**Value**
Simulated area	$800 \times 800\ m^2$
Number of forwarding nodes	20, 40, 60, 80, 100
AoI radius	50 m
Transmission range	100 m
Energy capacity	$3000 - 19000$ Joules
Energy model	`WifiRadioEnergyModel`, `BasicEnergySource`
Mobility model of forwarding nodes	`TruncatedLevyWalkModel`
#Operations in a processing task	5
#Operations available on each device	1, 2, 3, 4, 5
Simulation time	7-8 hours

devices in the traces make a long pause during the movement, such that their average speed is around 0.05 m/s. Without considering the pause time, the average moving speed of the simulated device is on average 0.15 m/s. Some outliers in the trace reach up to 6 m/s. Therefore, the devices with high speed can serve as mobile message ferries to bring the interest and data packets faster to the destination. However, since the Levy walk mobility traces are much less mobile compared to the mobility model used in Section 3.4, we expect slower time performance to all interest and data forwarding approaches overall.

For the evaluation of distributed data processing, we assume a full-collaborative environment among devices, such that each device always executes as many operations in the processing task of a data packet as possible if it possesses matching capabilities. Each data packet consists of 5 operations, which are set up in a random order. According to the definition of a complex task in distributed data processing, the operations have to be executed in the exact order which is specified in the naming scheme. We vary the number of available operations on each device from 1 to 5 operations.

The most important parameters used for the consolidated evaluation are summarized in Table 10. The parameters relevant for the newly added components in the simulation are highlighted.

In the following, we present the evaluation results for all three steps of information retrieval workflow. Note that, unsuccessful delivery in the measurements with "not assigned" values are omitted. We implemented and compared four forwarding approaches. These are (i) two-phase interest forwarding in combination with results



(a) All network densities

(b) Zoomed in at 60, 80, and 100 forwarding nodes

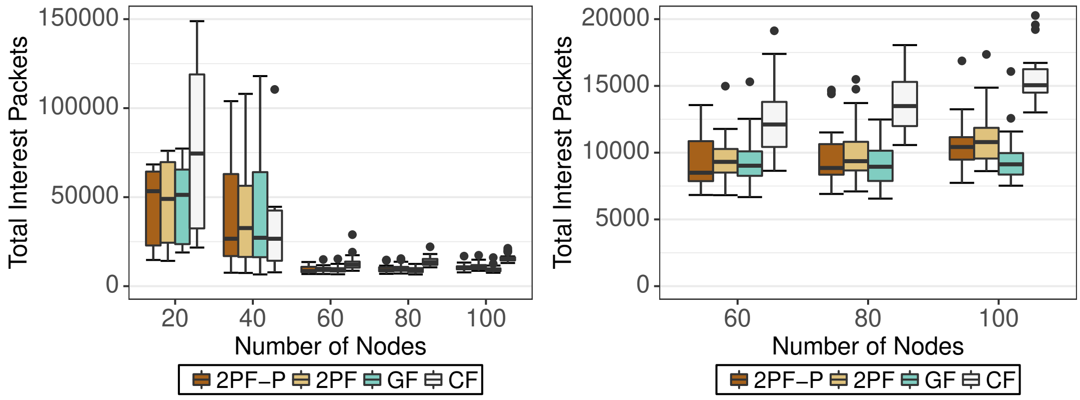
Figure 30: Time to find producer with Levy walk mobility model

delivery based on mobility prediction as introduced in this chapter (abbr. as *2PF-P*), (ii) two-phase interest forwarding in combination with controlled data flooding for results delivery as introduced in Chapter 3 (abbr. as *2PF*), (iii) distance-based geo-forwarding (abbr. as *GF*), and (iv) controlled flooding (abbr. as *CF*). *GF* and *CF* were introduced in Section 3.4. All results are reported with 95% confidence interval.

5.4.2 *Interest Forwarding Revisited*

In this section, we reevaluate the performance of interest forwarding using the Levy walk mobility model to assess the limitation of our proposed concepts. For this evaluation, we reuse two performance metrics that primarily characterize the performance of interest forwarding phase, i.e., *time to find producer* and *overhead* measured by the total number of interest packets. Additionally, since the Levy walk mobility model resembles human mobility patterns, we also introduce the accuracy metric for interest forwarding. The accuracy for the interest forwarding phase is calculated as the ratio between the number interest packets that are received by the information producers and the number of interest packets that are received by all devices in the system. Accordingly, the accuracy metric for interest forwarding augments the overhead measurement. More accurate interest forwarding implies that more interest packets as produced from the communication overhead are "goodput" for the transmissions, since they can reach information producers to trigger (new) data collection.

The results with regards to *time to find producers* and *interest packets overhead* are shown in Figure 30 and Figure 31, respectively. In both figures, we present the results measured for all network densities (from 20 to 100) and the same results focusing on dense networks (with 60, 80, 100 devices) with downscaled values to compare the performances among forwarding approaches more precisely. With regards to *time to find producer* the results in Figure 30 show that a sparse network yields a very slow *time to find producer*. With 20 forwarding devices, all forwarding approaches take on



(a) All network densities

(b) Zoomed in at 60, 80, and 100 forwarding nodes

Figure 31: Number of packets overhead generated during the interest forwarding phase with Levy walk mobility model

average 20 s to find the first mobile producer. With 40 forwarding devices, the time to find producer is down to an average of 5 s for all forwarding approaches. In a sparse network, the time to find producer fluctuates a lot regardless of the forwarding approaches, since only very few devices move fast enough to bring the interest packets to their destination. In a dense network having 60 to 100 forwarding devices, the time to find producer is improved drastically, down to less than 1 s for all approaches. The aforementioned results confirm, that the mobility model does affect the performance of crowd sensing tasks distribution. The low performance of slowly moving devices is compensated through a high number of devices in a dense opportunistic network. The downscaled results in Figure 30b show that our two-phase forwarding approach and geo-forwarding outperform controlled flooding in dense networks. Both two-phase forwarding and geo-forwarding require less than 25 ms to find mobile producers, while a controlled flooding takes up to average 500 ms. Thereby, the performances of two-phase forwarding and geo-forwarding are quite similar, since the devices in the simulation move considerably slower compared to the evaluation in Section 3.4. Slowly moving devices result in a higher chance to reach mobile producers within the AoI, since the interests can stay longer near the AoI. This observation hints at more suitability of our approach for a highly mobile scenario, while for slowly moving devices geo-forwarding without replication inside the buffer zone is sufficient and can save overhead generated by replicated rebroadcasting.

With regards to interest packet overhead, forwarding devices in a sparse network generates a lot of interest packets. This is due to the fact, that slowly moving devices in a sparse network can result into more packet loss; consequently leading to more retransmission. In contrast, this overhead is reduced in a dense network, since interest forwarding in a dense network requires less time to reach mobile producers. While the interest packets overhead generated by our two-phase forwarding approach and geo-forwarding are quite stable on average 10000 interest packets with 60, 80, 100 devices, the interest packets overhead generated by controlled flooding increases linearly. With

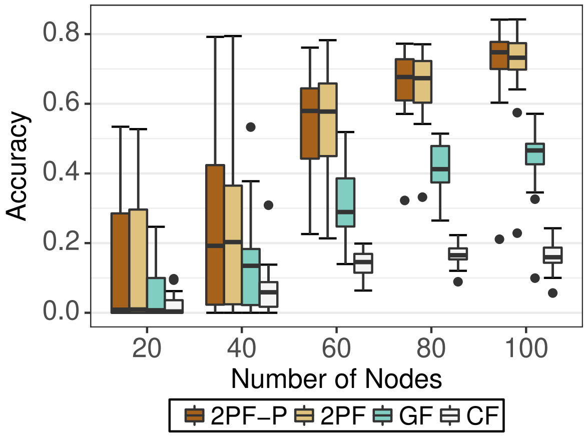


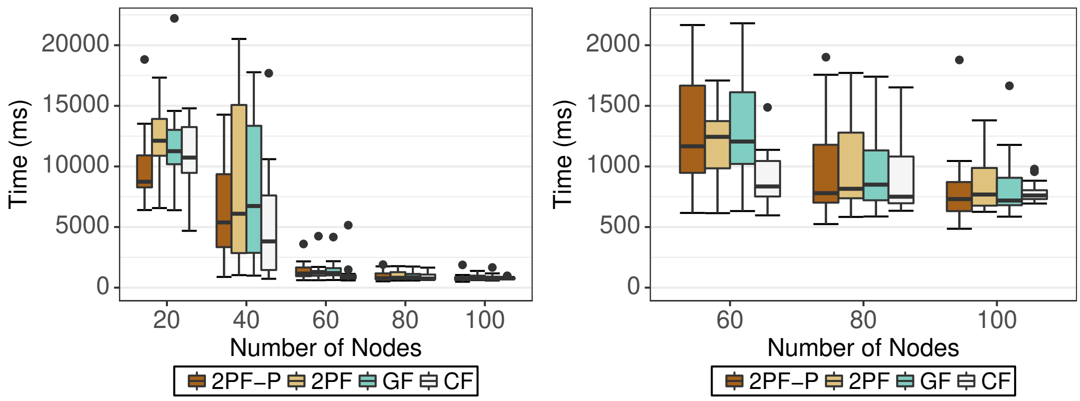
Figure 32: Accuracy of interest forwarding phase

100 forwarding devices, controlled flooding generates on average 15000 packets with several outliers even up to 20000 packets. Altogether, even in a simulation scenario with slowly moving devices the two-phase forwarding approach and geo-forwarding achieve much less time to find producer, while generating far less interest packets overhead.

The results for accuracy measure with regards to interest packets received by information producers are shown in Figure 32. Per our definition, the accuracy characterizes the "goodput" of interest packets transmission. We observe that the accuracy for a sparse network with 20 and 40 devices is on average lower than 20% for all forwarding approaches. However, several outliers show that our two-phase forwarding approach can still reach up to 80% accuracy depending on the movement traces. It is evident in Figure 32, that the accuracy for interest forwarding using our proposed two-phase forwarding approach increases for denser networks. In a dense network, our two-phase forwarding approach is much more accurate compared to geo-forwarding and controlled flooding. With 100 forwarding devices, the accuracy of two-phase forwarding lies on average near 80%, implying that 80% of the interest packets broadcast in the network can reach mobile information producers inside the AoI successfully. The accuracy of geo-forwarding only reaches on average 50% and of the accuracy of controlled flooding lies very low, less than 20%. This result shows that in order to reach mobile information producers, geo-forwarding and controlled flooding have to trade-off much communication overhead. Overall, the advantage of our two-phase forwarding approach to distribute crowd sensing tasks to the *appropriate* mobile information producers is confirmed.

5.4.3 *Data Forwarding*

In this section, we present the evaluation results of data forwarding as results delivery. We compare the performance of four data forwarding approaches *2PF-P, 2PF, GC, CF* with regards to *end to end delay* as results delivery time, *communication* overhead in terms of generated data packets, and accuracy of the data forwarding phase.



(a) All network densities

(b) Zoomed in at 60, 80, and 100 forwarding nodes

Figure 33: End to end delay with Levy Walk mobility model

Results *delivery time* is determined as the elapsed time after a mobile information consumer propagates the first interest packet until it receives the data packet. The results for end to end delay measures are shown in Figure 33. The results for number of data packets as overhead are shown in Figure 34. With regards to *delivery time*, we observe similar degrading time performance with the Levy walk mobility model compared to the evaluation with faster moving devices in Section 3.4. All forwarding approaches in sparse network with 20 and 40 devices take on average long time to deliver the results to mobile consumers. Regardless of data forwarding approaches, the results delivery is only successful with 60% of test runs for 20 forwarding devices and around 80% of test runs for 40 forwarding devices. In dense networks with more than 60 forwarding devices, the results delivery always succeeds. Consequently, the average deliver time measured in sparse networks with 20, 40 forwarding devices are meant to demonstrate the huge performance gap between sparse and dense networks. The main reason for this gap is due to slowly moving forwarding devices in the simulation scenario, therefore the performance can only be compensated with a high number of forwarding devices. In a dense network with 100 forwarding devices, our data forwarding approach based on mobility prediction for mobile consumers (2PF-P) achieves on average 750 ms, comparable to flooding-based broadcast which is utilized by other approaches. Thereby, controlled flooding is shown to provide more stable *results delivery time* with the highest density at 100 forwarding devices. However, flooding-based broadcast of data packets has to generate much (redundant) communication overhead in order to achieve results delivery.

We can observe from Figure 34, that the overhead generated by results delivery based on mobility prediction is much less compared to other approaches. With a dense network, the number of data packets generated by *2PF-P* increases from around 2500 packets at 60 devices up to 5000 packets at 100 devices. On the contrary, the number of generated data packets from *2PF* and *GF* increase from around 5000 at 60 devices

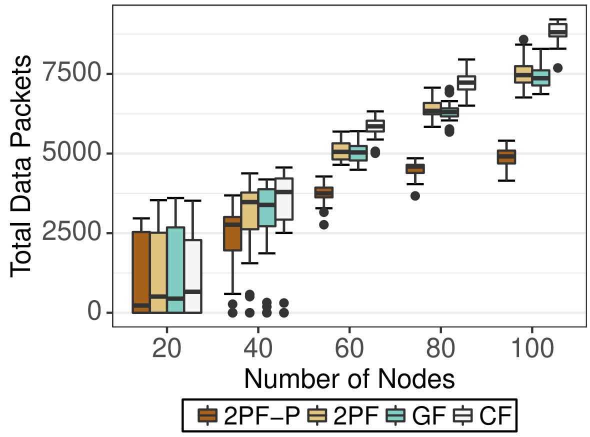


Figure 34: Number of packets as overhead generated during the data forwarding phase with Levy walk mobility model

up to 7500 at 100 devices. Controlled flooding generates the most overhead, around 6200 at 60 devices, up to around 9000 packets at 100 devices. The measurements confirm, that results delivery with mobility prediction can reduce overhead when delivering results to mobile consumers, assuming a dense opportunistic network. 2PF-P generates the least communication overhead, since in our data forwarding approach we only forward data packets into the direction of the predicted location. Thereby, the generated data packets are restricted. Overall, the results confirm the advantage of results delivery based on mobility prediction to achieve comparable delivery time compared to flooding-based data dissemination while generating much less overhead.

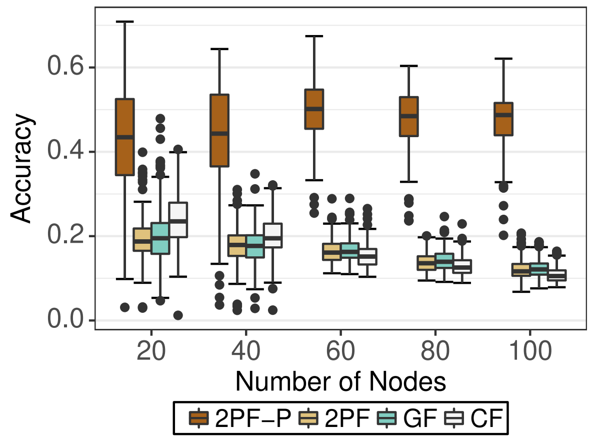
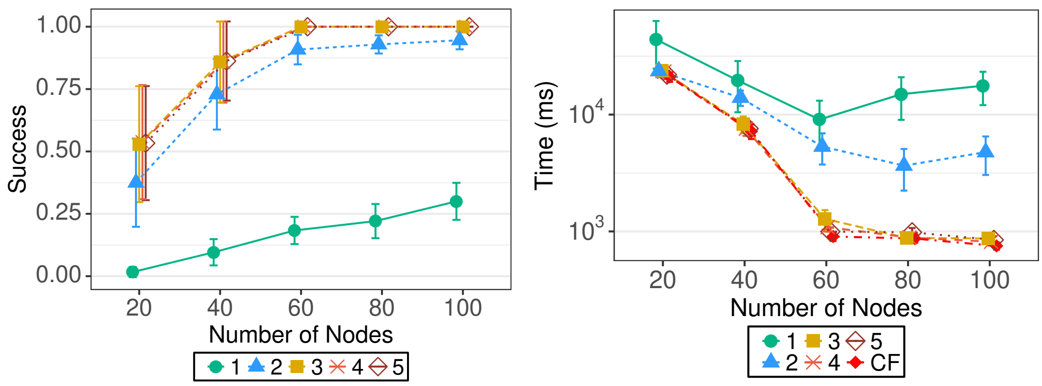


Figure 35: Accuracy of the data forwarding phase

We measure the accuracy for the data forwarding phase, defined similarly to the accuracy metric of interest forwarding. For data forwarding, the accuracy is determined as the ratio between the number of data packets that are received by the mobile information consumers and the number of data packets that are received by all devices in the network. The results for accuracy measurements are presented in Figure 35. The accuracy of 2PF-P fluctuates a lot in sparse networks, since in sparse networks less devices are located on the forwarding path leading to the predicted location. This can be improved in dense networks, showing less variance in the accuracy for results delivery with mobility prediction. On average, our proposed data forwarding based on mobility prediction achieves the best accuracy among all approaches. Regardless of network densities, the mobility prediction based approach can reach on average 50% accuracy, meaning 50% data packets of our data forwarding approach are "goodput". In contrast, all other approaches can achieve on average only around 20% accuracy. Together with the results for data packets overhead shown in Figure 34, the results show that our mobility prediction based approach is much more efficient than the others with regards to communication overhead reduction.

5.4.4 *Distributed In-Network Processing*

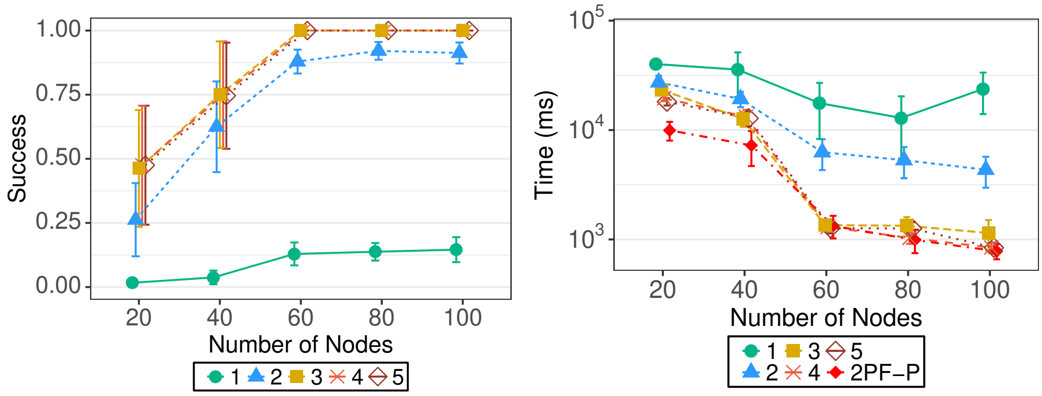
In this section, we present the evaluation results, when distributed data processing is integrated into the data forwarding phase. Thereby, the data packets are forwarded to the information consumers and are processed along the way by the forwarding devices which possess the matching capabilities. For this evaluation, each sensing task is attached with a processing task containing 5 operations in a random order. Each operation if being executed takes between 3 and 5 ms. Each forwarding device in the simulation executes as many operations as it can according to the available capabilities on the respective device. We use two evaluation metrics, i.e., the *success rate* of completing all processing tasks and the *delivery time* captured when the completely processed data is delivered. We integrate distributed data processing in controlled data flooding approach and our results delivery approach based on mobility prediction. The results are shown in Figure 36 and Figure 37.



(a) Success rate for distributed data processing with varying number of capabilities on each device

(b) Comparison of delivery time with and without processing. CF is without processing.

Figure 36: Distributed data processing integrated in results delivery using controlled flooding



(a) Success rate for distributed data processing with varying number of capabilities on each device

(b) Comparison of delivery time with and without processing. 2PF-P is without processing

Figure 37: Distributed data processing integrated in results delivery based on mobility prediction

The success rate of distributed data processing obtained in both flooding based data forwarding and mobility prediction based data forwarding show a similar trend. The success rate of the data processing is improved in two dimensions: (i) each forwarding device possesses more capabilities or (ii) the network has more forwarding devices. This trend is similar to the observation obtained in the evaluation of distributed processing using the task message template shown in Section 4.6. The success rate in a dense network having more than 60 forwarding devices, in which each device can execute at least 3 operations always reaches 100%. Comparing the results for the success rate in networks with 20 and 40 devices shown in Figure 36a and 37a, we can observe on average a better success rate when integrating distributed processing with controlled flooding. In such sparse networks distributed processing with controlled flooding achieves around 5% better success rate than distributed processing with results delivery based on mobility prediction. This effect is due to the fact that results delivery based on mobility prediction restricts the data forwarding region. Consequently, this approach restricts the chances to find capable operator devices likewise. However, as discussed previously the success rate can be compensated in dense networks and with a high number of capabilities available on each forwarding device.

With regards to the delivery time, the results show a monotonic decreasing trend regardless of the density for networks in which each forwarding device is capable of executing more than 3 operations. Thereby, as shown in Figure 36b the results delivery time when integrating distributed data processing with controlled flooding is comparable (almost identical) to the delivery time obtained by controlled flooding without processing. With results delivery based on mobility prediction, the delivery time with distributed data processing with more capable forwarding devices also decreases as shown in Figure 37b. The impact of dense networks on delivery time (with and without distributed processing) can be observed and confirmed in both controlled flooding and results delivery based on mobility prediction. When each forwarding device in networks provides more than 3 operations, the delivery time is improved manifold. Here, the delivery time reduces from around 40 s in a sparse network with 20 devices, down to less than 1 s in a dense network with 100 devices.

It can be observed that the capabilities of the forwarding devices affect not only the success rate, but also the results delivery time. In networks, in which each forwarding device can only execute up to 2 operations, the results delivery time first decreases with more forwarding devices in sparse networks (20, 40, and 60 devices). However in this case, the results delivery time rises again in denser networks (with 80 and 100 devices). Figure 36b shows, that the results delivery time rises again starting at 80 devices, especially in case each forwarding device can execute only 1 operation. Similar results are shown in Figure 37b. This effect is explained by the fact that the success rate for network with only 1 operation available on each forwarding device is very low. Consequently, a lot of packets remain in the network during the whole simulation, potentially leading to packet drops when the packets queue is full or to collisions despite an avoidance scheme.

Overall, the results and discussions show that integrating distributed data processing during the forwarding phase is possible. Furthermore, the performance with regards to results delivery time can be preserved with a dense, highly capable, and collaborative network.

5.5 DISCUSSION

In this chapter, we conclude the information retrieval workflow by considering the results delivery problem. Results delivery is different from information dissemination in the sense, that results delivery targets individual information consumers. Consequently, the support from the state-of-the-art flooding-based approaches for dissemination is insufficient for results delivery. Furthermore, mobile consumers and uncertainty of opportunistic networks make it more challenging to deliver results. To this end, we utilize mobility prediction in results delivery. Thereby, we rely on distributed shared context information and on autonomous decision of participating devices to make autonomous forwarding decisions. We show that results delivery based on mobility prediction reduces communication overhead immensely, while not scarifying much delivery time. Furthermore, we also demonstrate the successful integration of distributed processing in data forwarding phase for results delivery. All in all, the distributed processing should be integrated in data forwarding phase in a dense, highly capable, and collaborative network to leverage the benefits of both results delivery and distributed in-network processing.

6 SUMMARY, CONCLUSIONS, AND OUTLOOK

In this thesis, we have proposed a *tasking* concept that utilizes resources of mobile devices in an opportunistic network to retrieve and process information. As a result, our *tasking* concept enables providing and ensuring QoS for applications and services which are built upon this type of network. We conclude our work by first summarizing the contents discussed throughout the thesis and highlighting the core contributions. Based on our contributions, we can draw conclusions and point out open issues as well as potential future work.

6.1 SUMMARY OF THE THESIS

In Chapter 1 we gave an overview of the potential for mobile computing, which finds its use in several application domains, such as in emergency response scenario and information centric (vehicular) networks. Thereby, we motivated *opportunistic resource utilization* as the basis to develop and provide *QoS* for applications/services on opportunistic networks and elaborated on the challenges for ensuring *QoS*. We discussed background information and related work on *opportunistic resource utilization* in Chapter 2. We divided our *tasking* concept in three core contributions (i) creation and assignment of information tasks, (ii) distributed in-network processing of information, and (iii) delivery of processed results. Accordingly, we also reviewed the state of the art for location-based forwarding in opportunistic networks, as well as NDN-based mobile networks, for distributed in-network processing, and for information dissemination, which are relevant for the research contributions. Based on the discussion and on the identified research gap, we presented the following contributions in our work.

6.1.1 *Contributions*

As the first step of the *information retrieval workflow*, we proposed and presented a decentralized *tasking* concept [131] for crowd sensing task distribution in Chapter 3. Thereby, the crowd sensing tasks are injected, disseminated in opportunistic networks, and assigned to the appropriate mobile sensors, without having to rely on any centralized coordination/management. We based the *tasking model* for crowd sensing on the *NDN* paradigm, which focuses on addressing information/data, instead of addressing hosts. Within our *NDN*-based tasking model for crowd sensing request distribution, we proposed a naming scheme, that allows the *information consumers* to specify the granularity for quality of requested information. Based on the proposed naming scheme, we designed the *two-phase* interest forwarding approach, utilizing *self-organizing patterns* to guide the crowd sensing request in form of *NDN* interest packets towards appropriate *information producers*. To achieve distributed coordination, we embedded context information as attribute tags in each interest packet broadcast in the network. Extracting context information from attribute tags of an interest packet allows each device to make forwarding decisions autonomously. The context information used for interest forwarding is utilized not only to fulfill the quality requirements for crowd sensing requests, but also to reduce communication overhead and to balance resource consumption for forwarding. Existing approaches to location-based forwarding are designed with focus in messages delivery, ignoring the quality requirements of sensing task. Additionally, opportunistic network based crowd sensing application assume a rather homogenous setup. State of the art in interest forwarding in NDN-based network still has not fully addressed the heterogeneity of mobile *information producers*. In contrast, our *two-phase* forwarding approach, in combination with the proposed naming scheme allowed us to incorporate heterogeneity of devices, while fulfilling the quality requirements.

In the second step of the *information retrieval workflow*, having the (raw) data collected through crowd-based sensors, we investigated on the next *tasking* concept, that leverages idle resource of participating mobile devices opportunistically for distributed data processing in Chapter 4. Considering the challenges of opportunistic networks, and in order to achieve fully decentralized-tasking concept, we also aimed at self-organizing and autonomous decision for processing, while we still allowed for distributed coordination. As a result, we generalized and designed a technique to enable distributed processing, named *adaptive task-oriented message template* [134]. This task message template is self-encapsulated, allows to define and divide complex processing goals in several operations, together with the corresponding payload data attached. We achieved distributed processing on opportunistic network by passing the task messages from one device to another. During this task handover process, each device can make autonomous decision on how to contribute to the processing tasks. Thereby, we were able to not only utilize the idle resource, but also be able to leverage heterogeneous capabilities of the participating devices. The *tasking* concept based on tasks message template also allows us to derive handover strategies [136], using only local context information, to fulfill quality requirements for the processing tasks. Existing approaches in opportunistic computing still does not fully enable autonomous decision of participating devices in distributed processing.

Finally, in the last step of the *information retrieval workflow*, we proposed the integrated form of distributed processing in opportunistic networks, and results delivery, considering the mobility of the information requesters as the main driver in Chapter 5. Thereby, we first showed the integration of the processing task template concept into a NDN based data forwarding phase. Data packets are essential for a NDN based results delivery mechanism, since the measurements obtained as the results of interest forwarding will be encapsulated in this type of packet and broadcast in the whole network. Hence, we integrated a complex processing task into each data packet, so that the data in the packets can be processed by the forwarder devices on its way back to information consumers. Furthermore, to deliver the processed result to a mobile consumer, current approaches in NDN based mobile networks still mainly rely on broadcasting. However, broadcasting data in the whole opportunistic network, especially in resource-constraint scenarios such as in emergency situations, consumes storage and resource on participating devices. Therefore, we proposed to integrate mobility prediction for mobile information consumers, that allow the forwarder devices to focus the data forwarding in the predicted location of the consumers, thus saving resource for opportunistic network. We achieved distributed processing, and mobility prediction for information consumers, by again embedding context information into each interest as well as data packet.

As a result, we completed the whole *information retrieval workflow* on opportunistic networks, relying completely on distributed coordination and autonomous decision of participating devices.

6.1.2 *Conclusions*

We conducted evaluations for each step of the *information retrieval workflow*, which allowed us to analyze the performance, the ability to satisfy quality requirements in decentralized fashion, as well as to assess the overhead trade-off required for distributed coordination and autonomous decision of participating devices.

First, with regards to crowd sensing task distribution, we compared our *two-phase* forwarding approach against state of the art broadcasting in NDN-based mobile network, and geo-forwarding. We showed that our forwarding concept is able to find a mobile *information producer* faster, indicating the ability to find relevant data in a timely manner. However, to achieve timely search for the information producer, our *two-phase* forwarding approach had to rely on replicated interest broadcasting in the second phase, thus generated more overhead compared to geo-forwarding. Despite of that, due to the directive forwarding, and to the autonomous decision of forwarder devices to drop interest if the forwarder device cannot bring the interest closer to the requested region, the *two-phase* forwarding approach was still able to reduce the communication overhead manifold, compared to state of the art interest broadcasting in NDN-based mobile networks. Furthermore, we also showed that our approach was able to leverage context information embedded in interest packets to self-regulate the broadcasting rate, thus achieving fairer resource consumption among participating devices, with respect to forwarding contribution.

Second, with regards to distributed in-network processing concept for opportunistic networks, we compared the performance of our proposed tasks handover strategies, which rely only on local information and distributed coordination. With the results, we confirmed that local task handover mechanisms are able to achieve high success rate and high computation load balancing, low completion time, which are the main quality requirements for distributed processing. Again, we had to trade-off communication overhead for distributed coordination, in order to fulfill quality requirements. We could observe another finding from the evaluation results, that a single task handover strategy cannot cover all dimensions of quality for distributed processing. This observation serves as a motivation for adaptive transitions, to switch among different strategies during runtime as discussed in our future work.

Finally, to complete the *information retrieval workflow*, we conducted a consolidated evaluation for sensing task distribution, for the integration of tasks message template into the data packets of NDN based networks, and for the integration of mobility prediction for data forwarding as results delivery. We showed that, aligned with the observation of the generalization of our distributed processing concept, the data can be processed in decentralized fashion during the data forwarding phase. The success rate of this process could be improved when having more capabilities of devices available in the opportunistic networks. We also showed the time trade-off for having completely processed data at the information consumers, compared to normal data broadcasting. Last but not least, we showed a massive reduction of communication overhead, when leveraging mobility prediction for results delivery. The results of the consolidated evaluation also revealed and confirmed that mobility characteristics and network density have great impact on the performance of all decentralized concepts.

All in all, we showed that we were able to (i) allow distributed coordination by embedding context information in broadcast packets, (ii) leverage heterogeneity both in resource and capabilities for all steps of *information retrieval*, and (iii) fulfill fine-granular QoS requirements, relying only on distributed coordination.

6.2 OUTLOOK

To ensure the quality requirements of a task, we leverage the heterogeneous capabilities and resources provided by the devices in an opportunistic network. Therefore, we need to rely on the collaboration and cooperation of participating devices. We considered the emergency response as the motivating scenario, in which a trusted collaboration among devices can be assumed, since all devices share a common goal of conducting relief operations, hence benefit from the results. For generic applications of opportunistic networks, our contributions should be augmented by incentive mechanisms to motivate participating devices to *donate* computation and networking resources, as well as by security mechanisms to counter related issues which are inherent in opportunistic networks [95]. With regards to the incentive for participating devices, decentralized resource reservation scheme such as [94] can be utilized to allow devices that contribute more resources to reserve for more energy recharging in emergency communication networks. The monetary incentive can also be provided in a decentralized manner based on the performance and contribution of a device towards the completion of the task; to this end, a contract-based mechanism for a device-to-device communication such as [220] is worth being studied. With regards to security in opportunistic networks, Lilien et al. [95] pointed out several issues, which are also relevant for our work, e.g., providing false information/results which affects the quality of task execution, intentionally dropping packets/messages which worsens the communication quality. One approach to address these issues is to generate redundant messages/packets to circumvent behaviors of malicious devices. However, as a consequence, redundancy can negatively affect other quality requirements. We believe decentralized mechanisms to establish a trusted environment among devices [44] and to attest to the integrity of participating devices/software [81] are promising as future research directions. Furthermore, to protect the privacy of the participating devices in opportunistic networks, only information which is required to facilitate completing a task needs to be provided in our mechanisms. To enhance the privacy protection, the concept of *approximate computing* [65] can be utilized, in which task can be assigned to a group/a cluster of devices based on geographical location instead of targeting individual devices, thus be able to hide the identity of the devices.

The concepts developed in this thesis are designed with a focus on opportunistic networks, but are not only restricted within the application domain for mobile devices. In the automotive sector, research on autonomous driving cars has attracted much attention. To support autonomous driving, the participating cars should be able to acquire relevant information [126] to decide on driving maneuvers. Especially, acquiring and sharing information through VANET communication allow the cars to adjust local driving behavior collaboratively and timely. Our *information retrieval* concepts can be well applied in this scenario. With more precise positioning systems, and the ability to determine the *most probable path* through a cloud-based *horizon* system [25], the application in automotive scenario will empower our information searching and results delivery concepts with more accuracy, making them more efficient.

Mobility of human carriers for the devices in opportunistic networks is the enabler for communication, and applications/services built upon this type of network. At the same time, mobility is also one of the biggest challenges to ensure quality. The results obtained from this thesis also revealed and confirmed the varying performance of the concepts, depending on evaluated mobility models. As such, more realistic mobility models are required for future research of mobile communication systems in general. For instance, projects SMARTER[1] and NICER[2] have developed solutions for maintaining communication using mobile devices in emergency situations. Thereby, to evaluate their contributions, they have organized field tests and testbed with test persons, robots, consequently, realistic mobility traces can be extracted from these tests. Simulative evaluation using realistic mobility traces in turn can reveal the potential weak link of the concepts when being deployed in practice for a particular application domain, such as in emergency response. The insight of such study can be used as input for planning in crowd steering [149], or in placement of Unmanned Aerial Vehicle (UAV) between partitioned opportunistic networks [110], to improve the performance of the deployed systems.

Last but not least, decentralized concepts relying on distributed coordination and autonomous decision can cope with uncertainty of the opportunistic networks, as shown in our thesis. However, it is also evident, that a single strategy cannot fulfill all QoS requirements, as these sometimes contradict each other. To deal with this, the concept of *transition* has been extensively researched within the collaborative center MAKI—Multi-Mechanisms Adaptation for the Future Internet[3] and can potentially be integrated. With regards to *information retrieval* in emergency response, local strategies for distributed processing can be switched among several options, to ensure that in the

1 http://smarter-projekt.de
2 http://nicer.network/
3 https://www.maki.tu-darmstadt.de

worst case scenario, the information consumers can always receive some information to facilitate the planing of relief operations. Hereby, the final goals of achieving completeness and timeliness, consequently the corresponding local decision strategies, can be switched between each other to adapt the system with respect to how critical the emergency situation develops.

Our contributions for tasking concepts in order to retrieve and process information while ensuring *QoS* requirements form a solid foundation for the development and creation of applications/services on mobile opportunistic networks.

ACKNOWLEDGMENTS

This work has been funded by the LOEWE initiative (Hessen, Germany) within the "NICER–Networked Infrastructureless Cooperation for Emergency Response" project and by the German Federal Ministry of Education and Research (BMBF) Software Campus project "OppEPM–Opportunistic Exploiting the Power of Mobile Devices" [01IS12054]

[1] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, et al. "ndnSIM: NDN simulator for NS-3." In: *University of California, Los Angeles, Tech. Rep* 4 (2012).

[2] Vikas Agarwal, Nilanjan Banerjee, Dipanjan Chakraborty, and Sumit Mittal. "USense–A Smartphone Middleware for Community Sensing." In: *IEEE Mobile Data Management (MDM)*. Vol. 1. 2013, pp. 56–65.

[3] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. "A survey of information-centric networking." In: *IEEE Communications Magazine* 50.7 (2012).

[4] Syed Hassan Ahmed, Safdar Hussain Bouk, Muhammad Azfar Yaqub, Dongkyun Kim, Houbing Song, and Jaime Lloret. "CODIE: Controlled Data and Interest Evaluation in Vehicular Named Data Networks." In: *IEEE Transactions on Vehicular Technology* 65.6 (2016), pp. 3954–3963.

[5] Ian F Akyildiz, Özgür B Akan, Chao Chen, Jian Fang, and Weilian Su. "InterPlaNetary Internet: state-of-the-art and research challenges." In: *Computer Networks* 43.2 (2003), pp. 75–112.

[6] Ali M Alakeel. "A Guide to Dynamic Load Balancing in Distributed Computer Systems." In: *International Journal of Computer Science and Information Security* 10.6 (2010), pp. 153–160.

[7] Daniele Alessandrelli, Matteo Petraccay, and Paolo Pagano. "T-RES: Enabling Reconfigurable In-network Processing in IoT-based WSNs." In: *9th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2013, pp. 337–344.

[8] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation." In: *Journal of Network and Computer Applications* 50 (2015), pp. 148–158.

[9] Marica Amadeo, Antonella Molinaro, and Giuseppe Ruggeri. "E-CHANET: Routing, forwarding and transport in Information-Centric Multihop Wireless Networks." In: *Computer communications* 36.7 (2013), pp. 792–803.

[10] Nils Aschenbruck, Raphael Ernst, Elmar G.-Padilla, and Matthias Schwamborn. "BonnMotion: A Mobility Scenario Generation and Analysis Tool." In: *SIMUTools*. 2010.

[11] Priyanka Asrani. "Mobile cloud computing." In: *International Journal of Engineering and Advanced Technology* 2.4 (2013), pp. 606–609.

[12] Paolo Baronti, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu. "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards." In: *Computer communications* 30.7 (2007), pp. 1655–1695.

[13] Lars Baumgärtner, Pablo Graubner, Jonas Höchst, Anja Klein, and Bernd Freisleben. “Speak less, hear enough: On dynamic announcement intervals in wireless on-demand networks.” In: *Wireless On-demand Network Systems and Services (WONS)*. 2017, pp. 33–40.

[14] Divyashri Bhat, Cong Wang, Amr Rizk, and Michael Zink. “A load balancing approach for adaptive bitrate streaming in information centric networks.” In: *IEEE Multimedia & Expo Workshops (ICMEW)*. 2015, pp. 1–6.

[15] 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.

[16] Robert D Blumofe and Charles E Leiserson. “Scheduling Multithreaded Computations by Work Stealing.” In: *Journal of the ACM (JACM)* 46.5 (1999), pp. 720–748.

[17] Chiara Boldrini, Marco Conti, Jacopo Jacopini, and Andrea Passarella. “Hibop: a history based routing protocol for opportunistic networks.” In: *IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2007, pp. 1–12.

[18] Chiara Boldrini, Marco Conti, and Andrea Passarella. “ContentPlace: social-aware data dissemination in opportunistic networks.” In: *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*. ACM. 2008, pp. 203–210.

[19] Chiara Boldrini and Andrea Passarella. “Data dissemination in opportunistic networks.” In: *Mobile Ad Hoc Networking: Cutting Edge Directions* (2013), pp. 453–490.

[20] Boris Jan Bonfils and Philippe Bonnet. “Adaptive and decentralized operator placement for in-network query processing.” In: *Telecommunication Systems* 26.2-4 (2004), pp. 389–409.

[21] Safdar Hussain Bouk, Syed Hassan Ahmed, Dongkyun Kim, and Houbing Song. “Named-data-networking-based ITS for smart cities.” In: *IEEE Communications Magazine* 55.1 (2017), pp. 105–111.

[22] Michael I Brownfield, Kaveh Mehrjoo, Almohonad S Fayez, and Nathaniel J Davis Iv. “Wireless sensor network energy-adaptive MAC protocol.” In: (2006).

[23] Per Nikolaj D Bukh and Raj Jain. *The Art of Computer Systems Performance Analysis, Techniques for experimental Design, Measurement, Simulation and Modeling*. 1992.

[24] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. “Maxprop: Routing for vehicle-based disruption-tolerant networks.” In: *25th IEEE International Conference on Computer Communications (INFOCOM)*. 2006, pp. 1–11.

[25] Daniel Burgstahler, Athiona Xhoga, Christoph Peusens, Martin Moebus, Doreen Boehnstedt, and Ralf Steinmetz. “RemoteHorizon.KOM: Dynamic Cloud-based eHorizon.” In: *Automotive meets Electronics (AME)*. 2016, pp. 1–6.

[26] Andrew T Campbell, Shane B Eisenman, Nicholas D Lane, Emiliano Miluzzo, and Ronald A Peterson. "People-centric Urban Sensing." In: *Proceedings of the 2nd annual international workshop on Wireless internet*. 2006.

[27] Yue Cao, Zhili Sun, Haitham Cruickshank, and Fang Yao. "Approach-and-Roam (AaR): A geographic Routing Scheme for Delay/Disruption Tolerant Networks." In: *IEEE transactions on Vehicular Technology* 63.1 (2014), pp. 266–281.

[28] Yue Cao, Zhili Sun, Ning Wang, Maryam Riaz, Haitham Cruickshank, and Xiulei Liu. "Geographic-based spray-and-relay (GSaR): an efficient routing scheme for DTNs." In: *IEEE Transactions on Vehicular Technology* 64.4 (2015), pp. 1548–1564.

[29] Valeria Cardellini, Mirko D'Angelo, Vincenzo Grassi, Moreno Marzolla, and Raffaela Mirandola. "A decentralized approach to network-aware service composition." In: *European Conference on Service-Oriented and Cloud Computing*. 2015, pp. 34–48.

[30] Valeria Cardellini, Vincenzo Grassi, Francesco Lo Presti, and Matteo Nardelli. "Optimal operator placement for distributed stream processing applications." In: *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*. ACM. 2016, pp. 69–80.

[31] Gilbert Cassar, Payam Barnaghi, and Klaus Moessner. "Probabilistic matchmaking methods for automated service discovery." In: *IEEE Transactions on Services Computing* 7.4 (2014), pp. 654–666.

[32] Georgios Chatzimilioudis, Nikos Mamoulis, and Dimitrios Gunopulos. "A distributed technique for dynamic operator placement in wireless sensor networks." In: *IEEE Mobile Data Management (MDM)*. 2010, pp. 167–176.

[33] Nanxi Chen, Nicolás Cardozo, and Siobhán Clarke. "Goal-driven service composition in mobile and pervasive computing." In: *IEEE Transactions on Services Computing* (2016).

[34] Marco Conti, Chiara Boldrini, Salil S Kanhere, Enzo Mingozzi, Elena Pagani, Pedro M Ruiz, and Mohamed Younis. "From MANET to People-centric Networking: Milestones and open Research Challenges." In: *Computer Communications* 71 (2015).

[35] Marco Conti and Mohan Kumar. "Opportunities in opportunistic computing." In: *Computer* 43.1 (2010).

[36] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. "MAUI: making smartphones last longer with code offload." In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. 2010, pp. 49–62.

[37] Tom De Wolf and Tom Holvoet. "Design patterns for decentralised coordination in self-organising emergent systems." In: *International Workshop on Engineering Self-Organising Applications*. 2006, pp. 28–49.

[38] Gang Deng, Xiaoming Xie, Li Shi, and Rere Li. "Hybrid Information Forwarding in VANETs through Named Data Networking." In: *Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2015.

[39] Sanjay K Dhurandher, Deepak K Sharma, Sahil Gupta, Isaac Woungang, and Mohammad S Obaidat. "Integration of fixed and mobile infrastructure for message passing in opportunistic networks." In: *Journal of networks* 10.12 (2015), pp. 642–657.

[40] Sanjay K Dhurandher, Deepak Kumar Sharma, Isaac Woungang, and Shruti Bhati. "HBPR: history based prediction for routing in infrastructure-less opportunistic networks." In: *Advanced Information Networking and Applications (AINA)*. IEEE. 2013, pp. 931–936.

[41] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. "A survey of mobile cloud computing: architecture, applications, and approaches." In: *Wireless communications and mobile computing* 13.18 (2013), pp. 1587–1611.

[42] Milica Pejanovic Djurivsic, Zhilbert Tafa, Goran Dimic, and Veljko Milutinovic. "A survey of military applications of wireless sensor networks." In: *Mediterranean Conference on Embedded Computing (MECO)*. IEEE. 2012, pp. 196–199.

[43] Michael Doering, Sven Lahde, Johannes Morgenroth, and Lars Wolf. "IBR-DTN: an efficient implementation for embedded systems." In: *Proceedings of the third ACM workshop on Challenged networks*. 2008, pp. 117–120.

[44] Rahul Dwarakanath, Boris Koldehofe, Yashas Bharadwaj, The An Binh Nguyen, David Eyers, and Ralf Steinmetz. "TrustCEP: Adopting a Trust-Based Approach for Distributed Complex Event Processing." In: *IEEE International Conference on Mobile Data Management (MDM)*. 2017, pp. 30–39.

[45] Rahul Dwarakanath, Boris Koldehofe, and Ralf Steinmetz. "Operator Migration for Distributed Complex Event Processing in Device-to-Device Based Networks." In: *Proceedings of the 3rd Workshop on Middleware for Context-Aware Applications in the IoT*. 2016, pp. 13–18.

[46] Derek L Eager, Edward D Lazowska, and John Zahorjan. "Adaptive Load Sharing in homogeneous Distributed Systems." In: *IEEE transactions on software engineering* 5 (1986), pp. 662–675.

[47] Philip Harrison Enslow. "What is a" distributed" data processing system?" In: *Computer* 11.1 (1978), pp. 13–21.

[48] Kevin Fall. "A delay-tolerant network architecture for challenged internets." In: *Conference on applications, technologies, architectures, and protocols for computer communications*. 2003, pp. 27–34.

[49] Kevin Fall, Keith L Scott, Scott C Burleigh, Leigh Torgerson, Adrian J Hooke, Howard S Weiss, Robert C Durst, and Vint Cerf. "Delay-tolerant networking architecture." In: *RFC*. 2007.

[50] Jialu Fan, Jiming Chen, Yuan Du, Ping Wang, and Youxian Sun. "Delque: A socially aware delegation query scheme in delay-tolerant networks." In: *IEEE Transactions on Vehicular Technology* 60.5 (2011), pp. 2181–2193.

[51] Karoly Farkas and Imre Lendak. "Simulation environment for investigating crowd-sensing based urban parking." In: *Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE. 2015, pp. 320–327.

[52] Niroshinie Fernando, Seng W Loke, and Wenny Rahayu. "Mobile Crowd Computing with Work Stealing." In: *IEEE NBiS*. 2012.

[53] Niroshinie Fernando, Seng W Loke, and Wenny Rahayu. "Mobile cloud computing: A survey." In: *Future generation computer systems* 29.1 (2013), pp. 84–106.

[54] Raghu K Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F Abdelzaher. "GreenGPS: a participatory sensing fuel-efficient maps application." In: *Proceedings of the international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 151–164.

[55] Raghu K Ganti, Fan Ye, and Hui Lei. "Mobile crowdsensing: current state and future challenges." In: *IEEE Communications Magazine* 49.11 (2011).

[56] Shuai Gao, Hongke Zhang, and Beichuan Zhang. "Energy efficient interest forwarding in NDN-based wireless sensor networks." In: *Mobile Information Systems* 2016 (2016).

[57] Paul Gardner-Stephen, Romana Challans, Jeremy Lakeman, Andrew Bettison, Dione Gardner-Stephen, and Matthew Lloyd. "The serval mesh: A platform for resilient communications in disaster & crisis." In: *IEEE Global Humanitarian Technology Conference (GHTC)*. 2013, pp. 162–166.

[58] Giulio Grassi, Davide Pesavento, Giovanni Pau, Lixia Zhang, and Serge Fdida. "Navigo: Interest forwarding by geolocations in vehicular named data networking." In: *IEEE World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2015, pp. 1–10.

[59] Pablo Graubner, Patrick Lampe, Jonas Höchst, Lars Baumgärtner, Mira Mezini, and Bernd Freisleben. "Opportunistic Named Functions in Disruption-tolerant Emergency Networks." In: *ACM International Conference on Computing Frontiers (CF)*. 2018.

[60] Christin Groba and Siobhan Clarke. "Opportunistic composition of sequentially-connected services in mobile computing environments." In: *IEEE Web Services (ICWS)*. 2011, pp. 17–24.

[61] Christin Groba and Siobhán Clarke. "Opportunistic service composition in dynamic ad hoc environments." In: *IEEE Transactions on Services Computing* 7.4 (2014), pp. 642–653.

[62] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang, and Xingshe Zhou. "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm." In: *ACM Computing Surveys (CSUR)* 48.1 (2015), p. 7.

[63] Bin Guo, Zhiwen Yu, Xingshe Zhou, and Daqing Zhang. "From participatory sensing to mobile crowd sensing." In: *IEEE Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 2014, pp. 593–598.

[64] Karim Habak, Mostafa Ammar, Khaled A Harras, and Ellen Zegura. "Femto clouds: Leveraging mobile devices to provide cloud service at the edge." In: *IEEE Cloud Computing (CLOUD)*. 2015, pp. 9–16.

[65] Jie Han and Michael Orshansky. "Approximate computing: An emerging paradigm for energy-efficient design." In: *Test Symposium (ETS), 2013 18th IEEE European*. IEEE. 2013, pp. 1–6.

[66] Ronny Hans, Daniel Burgstahler, Alexander Mueller, Manuel Zahn, and Dominik Stingl. "Knowledge for a longer life: development impetus for energy-efficient smartphone applications." In: *IEEE Mobile Services (MS)*. 2015, pp. 128–133.

[67] Daojing He, Sammy Chan, and Mohsen Guizani. "User privacy and data trustworthiness in mobile crowd sensing." In: *IEEE Wireless Communications* 22.1 (2015), pp. 28–34.

[68] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. "Network simulations with the ns-3 simulator." In: *SIGCOMM demonstration* 14.14 (2008), p. 527.

[69] Ivan Howitt and Jore A Gutierrez. "IEEE 802.15. 4 low rate-wireless personal area network coexistence issues." In: *IEEE Wireless Communications and Networking (WCNC)*. Vol. 3. 2003, pp. 1481–1486.

[70] Wei-Jen Hsu, Debojyoti Dutta, and Ahmed Helmy. "CSI: A paradigm for behavior-oriented profile-cast services in mobile networks." In: *Ad Hoc Networks* 10.8 (2012), pp. 1586–1602.

[71] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. "Profile-cast: Behavior-aware mobile networking." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2008, pp. 3033–3038.

[72] Haiming Jin, Lu Su, Danyang Chen, Klara Nahrstedt, and Jinhui Xu. "Quality of information aware incentive mechanisms for mobile crowd sensing systems." In: *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM. 2015, pp. 167–176.

[73] Hao Jin, Shidong Yan, Chenglin Zhao, and Dong Liang. "PMC2O: Mobile cloudlet networking and performance analysis based on computation offloading." In: *Ad Hoc Networks* 58 (2017), pp. 86–98.

[74] Salil S Kanhere. "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces." In: *IEEE Mobile Data Management (MDM)*. Vol. 2. 2011, pp. 3–6.

[75] Merkouris Karaliopoulos, Orestis Telelis, and Iordanis Koutsopoulos. "User recruitment for mobile crowdsensing over opportunistic networks." In: *IEEE International Conference on Computer Communications (INFOCOM)*. 2015, pp. 2254–2262.

[76] Bhanu Kaushik, Honggang Zhang, Xinwen Fu, Benyuan Liu, and Jie Wang. "Smartparcel: A collaborative data sharing framework for mobile operating systems." In: *IEEE Distributed Computing Systems Workshops (ICDCSW)*. 2013, pp. 290–295.

[77] Ryoma Kawajiri, Masamichi Shimosaka, and Hisashi Kashima. "Steered crowdsensing: Incentive design towards quality-oriented place-centric crowdsensing." In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2014, pp. 691–701.

[78] Roelof Kemp, Nicholas Palmer, Thilo Kielmann, and Henri Bal. "Cuckoo: a computation offloading framework for smartphones." In: *International Conference on Mobile Computing, Applications, and Services*. Springer. 2010, pp. 59–79.

[79] Maurice J Khabbaz, Chadi M Assi, and Wissam F Fawaz. "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges." In: *IEEE Communications Surveys & Tutorials* 14.2 (2012), pp. 607–640.

[80] Young-Bae Ko and Nitin H Vaidya. "Location-Aided Routing (LAR) in mobile ad hoc networks." In: *Wireless networks* 6.4 (2000), pp. 307–321.

[81] Florian Kohnhäuser, Niklas Büscher, and Stefan Katzenbeisser. "SALAD: Secure and Lightweight Attestation of Highly Dynamic and Disruptive Networks." In: *Proceedings of the Asia Conference on Computer and Communications Security*. ASIACCS. 2018.

[82] Meng Kuai, Xiaoyan Hong, and Qiangyuan Yu. "Density-Aware Delay-Tolerant Interest Forwarding in Vehicular Named Data Networking." In: *Vehicular Technology Conference (VTC)*. 2016.

[83] Karthik Kumar and Yung-Hsiang Lu. "Cloud computing for mobile users: Can offloading computation save energy?" In: *Computer* 43.4 (2010), pp. 51–56.

[84] Patrick Lampe, Lars Baumgärtner, Ralf Steinmetz, and Bernd Freisleben. "SmartFace: Efficient face detection on smartphones for wireless on-demand emergency networks." In: *IEEE Telecommunications (ICT)*. 2017, pp. 1–7.

[85] Nicholas D Lane, Shane B Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T Campbell. "Urban sensing systems: opportunistic or participatory?" In: *Proceedings of the 9th workshop on Mobile computing systems and applications*. ACM. 2008, pp. 11–16.

[86] Ray R Larson. “Introduction to information retrieval.” In: *Journal of the American Society for Information Science and Technology* 61.4 (2010), pp. 852–853.

[87] Jason LeBrun, Chen-Nee Chuah, Dipak Ghosal, and Michael Zhang. “Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks.” In: *IEEE Vehicular technology conference (VTC)*. Vol. 4. 2005, pp. 2289–2293.

[88] Uichin Lee, Jiyeon Lee, Joon-Sang Park, and Mario Gerla. “FleaNet: A virtual market place on vehicular networks.” In: *IEEE Transactions on Vehicular Technology* 59.1 (2010), pp. 344–355.

[89] Hanshang Li, Ting Li, and Yu Wang. “Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks.” In: *IEEE Mobile Ad Hoc and Sensor Systems (MASS)*. 2015, pp. 136–144.

[90] Peng Li, Tao Zhang, Chuanhe Huang, Xi Chen, and Bin Fu. “RSU-Assisted Geocast in Vehicular Ad Hoc Networks.” In: *IEEE Wireless Communications* 24.1 (2017), pp. 53–59.

[91] Qingyu Li, Panlong Yang, Yubo Yan, and Yue Tao. “Your friends are more powerful than you: Efficient task offloading through social contacts.” In: *IEEE Communications (ICC)*. 2014, pp. 88–93.

[92] Yong Li, Yurong Jiang, Depeng Jin, Li Su, Lieguang Zeng, and Dapeng Wu. “Energy-efficient optimal opportunistic forwarding for delay-tolerant networks.” In: *IEEE Transactions on Vehicular Technology* 59.9 (2010), pp. 4500–4512.

[93] Patrick Lieser, Flor Alvarez, Paul Gardner-Stephen, Matthias Hollick, and Doreen Boehnstedt. “Architecture for Responsive Emergency Communications Networks.” In: *Global Humanitarian Technology Conference (GHTC)*. IEEE. 2017.

[94] Patrick Lieser, Nils Richerzhagen, Tim Feuerbach, Leonhard Nobach, Doreen Böhnstedt, and Ralf Steinmetz. “Take it or Leave it: Decentralized Resource Allocation in Mobile Networks.” In: *42nd IEEE Conference on Local Computer Networks (LCN)*. 2017.

[95] Leszek Lilien, Zille Huma Kamal, Vijay Bhuse, and Ajay Gupta. “Opportunistic Networks: The Concept and Research Challenges in Privacy and Security.” In: *Proc. of the WSPWN* (2006), pp. 134–147.

[96] Leszek Lilien, Ajay Gupta, Zijiang Yang, et al. “Opportunistic resource utilization networks—a new paradigm for specialized ad hoc networks.” In: *Computers & electrical engineering* 36.2 (2010), pp. 328–340.

[97] Yujin Lim, Sanghyun Ahn, and Kwon-Hee Cho. “Abiding geocast for commercial ad dissemination in the vehicular ad hoc network.” In: *IEEE Consumer Electronics (ICCE)*. 2011, pp. 115–116.

[98] Anders Lindgren, Avri Doria, and Olov Schelen. “Probabilistic Routing in Intermittently Connected Cetworks.” In: *Service assurance with partial and intermittent resources*. 2004, pp. 239–254.

[99] Anders Lindgren and Pan Hui. "The Quest for a Killer App for opportunistic and Delay Tolerant Networks." In: *Proceedings of the 4th ACM workshop on Challenged networks*. ACM. 2009.

[100] Chi Harold Liu, Bo Zhang, Xin Su, Jian Ma, Wendong Wang, and Kin K Leung. "Energy-aware participant selection for smartphone-enabled mobile crowd sensing." In: *IEEE Systems Journal* 11.3 (2017), pp. 1435–1446.

[101] Xuan Liu, Zhuo Li, Peng Yang, and Yongqiang Dong. "Information-centric mobile ad hoc networks and content routing: a survey." In: *Ad Hoc Networks* 58 (2017), pp. 255–268.

[102] Yan Liu, Bin Guo, Yang Wang, Wenle Wu, Zhiwen Yu, and Daqing Zhang. "TaskMe: multi-task allocation in mobile crowd sensing." In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016, pp. 403–414.

[103] Yang Liu, Zhipeng Yang, Ting Ning, and Hongyi Wu. "Efficient quality-of-service (QoS) support in mobile opportunistic networks." In: *IEEE Transactions on Vehicular Technology* 63.9 (2014), pp. 4574–4584.

[104] Christian Lochert, Björn Scheuermann, and Martin Mauve. "A Survey on Congestion Control for Mobile Ad Hoc Networks." In: *Wireless communications and mobile computing* 7.5 (2007), pp. 655–676.

[105] Konrad Lorincz, David J Malan, Thaddeus RF Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoffrey Mainland, Matt Welsh, and Steve Moulton. "Sensor networks for emergency response: challenges and opportunities." In: *IEEE Pervasive Computing* 3.4 (2004), pp. 16–23.

[106] Hong Lu, Nicholas D Lane, Shane B Eisenman, and Andrew T Campbell. "Bubble-sensing: Binding sensing tasks to the physical world." In: *Pervasive and Mobile Computing* 6.1 (2010), pp. 58–71.

[107] Shanshan Lu and Yonghe Liu. "Geoopp: Geocasting for opportunistic Networks." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2014, pp. 2582–2587.

[108] Thomas Ludwig, Christian Reuter, Tim Siebigteroth, and Volkmar Pipek. "CrowdMonitor: Mobile crowd sensing for assessing physical and digital activities of citizens during emergencies." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 4083–4092.

[109] Manisha Luthra, Boris Koldehofe, Pascal Weisenburger, Guido Salvaneschi, and Raheel Arif. "TCEP: Adapting to Dynamic User Environments by Enabling Transitions between Operator Placement Mechanisms (accepted for publication)." In: *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems (DEBS)*. June 2018.

[110] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. "Placement optimization of UAV-mounted mobile base stations." In: *IEEE Communications Letters* 21.3 (2017), pp. 604–607.

[111] Huadong Ma, Dong Zhao, and Peiyan Yuan. "Opportunities in mobile crowd sensing." In: *IEEE Communications Magazine* 52.8 (2014), pp. 29–35.

[112] Yaozhou Ma and Abbas Jamalipour. "Opportunistic geocast in disruption-tolerant networks." In: *IEEE Global Telecommunications Conference (GLOBECOM)*. 2011, pp. 1–5.

[113] Derek Hylton Maling. *Coordinate systems and map projections*. Elsevier, 2013.

[114] Yuyi Mao, Jun Zhang, and Khaled B Letaief. "Dynamic computation offloading for mobile-edge computing with energy harvesting devices." In: *IEEE Journal on Selected Areas in Communications* 34.12 (2016), pp. 3590–3605.

[115] Abraham Martin-Campillo, Jon Crowcroft, Eiko Yoneki, and Ramon Martin. "Evaluating Opportunistic Networks in Disaster Scenarios." In: *Journal of Network and computer applications* 36.2 (2013), pp. 870–880.

[116] Davide Mascitti, Marco Conti, Andrea Passarella, and Laura Ricci. "Service provisioning through opportunistic computing in mobile clouds." In: *Procedia Computer Science* 40 (2014), pp. 143–150.

[117] Yasser Mawad and Stefan Fischer. "Infrastructure-based delay tolerant network communication." In: *International Conference on Information Networking (ICOIN)*. 2016, pp. 161–165.

[118] Michael Meisel, Vasileios Pappas, and Lixia Zhang. "Listen first, broadcast later: Topology-agnostic forwarding under high dynamics." In: *Annual conference of international technology alliance in network and information science*. 2010, p. 8.

[119] C. Meurisch, J. Gedeon, T. A. B. Nguyen, F. Kaup, and M. Muhlhauser. "Decision Support for Computational Offloading by Probing Unknown Services." In: *International Conference on Computer Communication and Networks (ICCCN)*. 2017, pp. 1–9.

[120] Christian Meurisch, Julien Gedeon, Artur Gogel, The An Binh Nguyen, Fabian Kaup, Florian Kohnhaeuser, Lars Baumgaertner, Milan Schmittner, and Max Muehlhaeuser. "Temporal Coverage Analysis of Router-Based Cloudlets Using Human Mobility Patterns." In: *IEEE Global Communications Conference (GLOBECOM)*. Dec. 2017, pp. 1–6.

[121] Christian Meurisch, The An Binh Nguyen, Julien Gedeon, Florian Konhauser, Milan Schmittner, Stefan Niemczyk, Stefan Wullkotte, and Max Muhlhauser. "Upgrading Wireless Home Routers as Emergency Cloudlet and Secure DTN Communication Bridge." In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–2.

[122] Christian Meurisch, Alexander Seeliger, Benedikt Schmidt, Immanuel Schweizer, Fabian Kaup, and Max Mühlhäuser. "Upgrading wireless home routers for enabling large-scale deployment of cloudlets." In: *International Conference on Mobile Computing, Applications, and Services*. 2015, pp. 12–29.

[123] Christian Meurisch, Ashwinkumar Yakkundimath, Benedikt Schmidt, and Max Mühlhäuser. "Upgrading Wireless Home Routers as Emergency Cloudlet: A Runtime Measurement." In: *Mobile Computing, Applications, and Services* (2015), p. 338.

[124] Tobias Meuser, Patrick Lieser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "Adaptive Information Aggregation for Application-specific Demands." In: *Proceedings of the 1st International Balkan Conference on Communications and Networking*. May 2017.

[125] Tobias Meuser, Björn Richerzhagen, Ioannis Stavrakakis, The An Binh Nguyen, and Ralf Steinmetz. "Relevance-Aware Information Dissemination in Vehicular Networks." In: *Proceedings of IEEE WoWMoM*. 2018.

[126] Tobias Meuser, Martin Wende, Patrick Lieser, Björn Richerzhagen, and Ralf Steinmetz. "Adaptive Decision Making Based on Temporal Information Dynamics." In: *Vehicle Technology and Intelligent Transport Systems (VEHITS)*. 2018.

[127] Waldir Moreira and Paulo Mendes. "Pervasive data sharing as an enabler for mobile citizen sensing systems." In: *IEEE Communications Magazine* 53.10 (2015), pp. 164–170.

[128] Abderrahmen Mtibaa, Afnan Fahim, Khaled A Harras, and Mostafa H Ammar. "Towards resource sharing in mobile device clouds: Power balancing across mobile devices." In: *ACM SIGCOMM Computer Communication Review*. Vol. 43. 4. ACM. 2013, pp. 51–56.

[129] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Peter Boda. "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research." In: *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM. 2009, pp. 55–68.

[130] Mirco Musolesi and Cecilia Mascolo. "CAR: Context-aware adaptive routing for delay-tolerant mobile networks." In: *IEEE Transactions on Mobile Computing* 8.2 (2009), pp. 246–260.

[131] The An Binh Nguyen, Pratyush Agnihotri, Christian Meurisch, Manisha Luthra, Rahul Dwarakanath, Jeremias Blendin, Doreen Böhnstedt, Michael Zink, and Ralf Steinmetz. "Efficient Crowd Sensing Task Distribution Through Context-aware NDN-based Geocast." In: *42nd Conference on Local Computer Networks (LCN)*. Oct. 2017, pp. 52–60.

[132] The An Binh Nguyen, Frank Englert, Simon Farr, Christian Gottron, Doreen Böhnstedt, and Ralf Steinmetz. "Hybrid Communication Architecture for Emergency Response - An Implementation in Firefighter's Use Case." In: *Proceedings of the fifth International Workshop on Pervasive Networks for Emergency Management (PerNEM)*. Mar. 2015, pp. 528–533.

[133] The An Binh Nguyen, Christian Klos, Christian Meurisch, Patrick Lampe, Björn Richerzhagen, and Ralf Steinmetz. "Demo: Enabling In-Network Processing utilizing Nearby Device-to-Device Communication." In: *IFIP Networking*. May 2018, A7–A8.

[134] The An Binh Nguyen, Christian Meurisch, Stefan Niemczyk, Doreen Böhnstedt, Kurt Geihs, Max Mühlhäuser, and Ralf Steinmetz. "Adaptive Task-Oriented Message Template for In-Network Processing." In: *Proc. International Conference on Networked Systems (NetSys)*. Mar. 2017, pp. 1–8.

[135] The An Binh Nguyen, Christian Meurisch, Stefan Niemczyk, Christian Klos, Doreen Böhnstedt, and Ralf Steinmetz. "Demo: Facilitating Volunteer Computing Resources for In-Network Processing through Message Template." In: *International Conference on Networked Systems (NetSys)*. Mar. 2017, pp. 1–2.

[136] The An Binh Nguyen, Marius Rettberg-Päplow, Christian Meurisch, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. "Complex Services Offloading in Opportunistic Networks." In: *IFIP Networking*. May 2018, pp. 532–540.

[137] The An Binh Nguyen, Melanie Siebenhaar, Ronny Hans, and Ralf Steinmetz. "Role-based Templates for Cloud Monitoring." In: *IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC)*. Dec. 2014, pp. 242–250.

[138] Ory Okolloh. "Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information." In: *Participatory learning and action* 59 (2009), pp. 65–70.

[139] Ott, Jörg and Hyytiä, Esa and Lassila, Pasi and Vaegs, Tobias and Kangasharju, Jussi. "Floating content: Information sharing in urban areas." In: *Pervasive Computing and Communications (PerCom)*. IEEE. 2011.

[140] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, Kirak Hong, David Lillethun, and Umakishore Ramachandran. "MCEP: a mobility-aware complex event processing system." In: *ACM Transactions on Internet Technology (TOIT)* 14.1 (2014), p. 6.

[141] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, and Umakishore Ramachandran. "MigCEP: operator migration for mobility driven distributed complex event processing." In: *Proceedings of the 7th ACM international conference on Distributed event-based systems*. ACM. 2013, pp. 183–194.

[142] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. "Uncovering the overlapping community structure of complex networks in nature and society." In: *Nature* 435.7043 (2005), p. 814.

[143] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. "Crowd sensing of traffic anomalies based on human mobility and social media." In: *ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. 2013, pp. 344–353.

[144] Xiaoshan Pan, Charles S Han, Ken Dauber, and Kincho H Law. "A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations." In: *Ai & Society* 22.2 (2007), pp. 113–132.

[145] Sooksan Panichpapiboon and Wasan Pattara-Atikom. "A review of Information Dissemination protocols for Vehicular Ad Hoc Networks." In: *IEEE Communications Surveys & Tutorials* 14.3 (2012), pp. 784–798.

[146] Andrea Passarella, Mohan Kumar, Marco Conti, and Elenora Borgia. "Minimum-delay service provisioning in opportunistic networks." In: *IEEE Transactions on Parallel and Distributed Systems* 22.8 (2011), pp. 1267–1275.

[147] Luciana Pelusi, Andrea Passarella, and Marco Conti. "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks." In: *IEEE communications Magazine* 44.11 (2006).

[148] Davide Pesavento, Giulio Grassi, Claudio E Palazzi, and Giovanni Pau. "A naming scheme to represent geographic areas in NDN." In: *IFIP Wireless Days (WD)*. 2013, pp. 1–3.

[149] Danilo Pianini, Mirko Viroli, Franco Zambonelli, and Alois Ferscha. "HPC from a self-organisation perspective: the case of crowd steering at the urban scale." In: *IEEE High Performance Computing & Simulation (HPCS)*. 2014, pp. 460–467.

[150] Mikko Pitkänen, Teemu Kärkkäinen, Jörg Ott, Marco Conti, Andrea Passarella, Silvia Giordano, Daniele Puccinelli, Franck Legendre, Sacha Trifunovic, Karin Hummel, et al. "SCAMPI: service platform for social aware mobile and pervasive computing." In: *Workshop on Mobile Cloud Computing (MCC)*. ACM. 2012, pp. 7–12.

[151] Layla Pournajaf, Li Xiong, Vaidy Sunderam, and Slawomir Goryczka. "Spatial task assignment for crowd sensing with cloaked locations." In: *IEEE Mobile Data Management (MDM)*. Vol. 1. 2014, pp. 73–82.

[152] Rüdiger Pryss, Manfred Reichert, Jochen Herrmann, Berthold Langguth, and Winfried Schlee. "Mobile Crowd Sensing in Clinical and Psychological Trials–A Case Study." In: *IEEE Computer-Based Medical Systems (CBMS)*. 2015, pp. 23–24.

[153] Rüdiger Pryss, Manfred Reichert, Berthold Langguth, and Winfried Schlee. "Mobile crowd sensing services for tinnitus assessment, therapy, and research." In: *Mobile Services (MS)*. IEEE. 2015, pp. 352–359.

[154] Ioannis Psaras, Lorenzo Saino, Mayutan Arumaithurai, KK Ramakrishnan, and George Pavlou. "Name-based replication priorities in disaster cases." In: *IEEE Computer Communications Workshops (INFOCOM Workshop)*. IEEE. 2014, pp. 434–439.

[155] Moo-Ryong Ra, Bin Liu, Tom F La Porta, and Ramesh Govindan. "Medusa: A programming framework for crowd-sensing applications." In: *ACM Proceedings of the 10th international conference on Mobile systems, applications, and services*. 2012, pp. 337–350.

[156] Aydin Rajaei, Dan Chalmers, Ian Wakeman, and George Parisis. "GSAF: Efficient and Flexible Geocasting for Opportunistic Networks." In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2016, pp. 1–9.

[157] Sasank Reddy, Deborah Estrin, and Mani Srivastava. "Recruitment framework for participatory sensing data collections." In: *International Conference on Pervasive Computing*. Springer. 2010, pp. 138–155.

[158] Rana Asif Rehman and KIM Byung-Seo. "Location-Aware Forwarding and Caching in CCN-Based Mobile Ad Hoc Networks." In: *IEICE Transaction on Information and Systems* 99.5 (2016), pp. 1388–1391.

[159] Rana Asif Rehman, Tran Dinh Hieu, Hong-Min Bae, Sung-Hoon Mah, and Byung-Seo Kim. "Robust and efficient multipath Interest forwarding for NDN-based MANETs." In: *IEEE Wireless and Mobile Networking Conference (WMNC)*. 2016, pp. 187–192.

[160] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. "On the Levy-Walk Nature of Human Mobility." In: *IEEE/ACM transactions on networking (TON)* 19.3 (2011), pp. 630–643.

[161] Björn Richerzhagen, Nils Richerzhagen, Sophie Schonherr, Rhaban Hark, and Ralf Steinmetz. "Stateless Gateways-Reducing Cellular Traffic for Event Distribution in Mobile Social Applications." In: *IEEE Computer Communication and Networks (ICCCN)*. 2016, pp. 1–9.

[162] Nils Richerzhagen, Roland Kluge, Björn Richerzhagen, Patrick Lieser, Boris Koldehofe, Ioannis Stavrakakis, and Ralf Steinmetz. "Better Together: Collaborative Monitoring for Location-based Services." In: *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2018, pp. 1–9.

[163] Nils Richerzhagen, Björn Richerzhagen, Boris Koldehofe, and Ralf Steinmetz. "Towards Transitions Between Role Assignment Schemes: Enabling Adaptive Offloading in Challenged Networks." In: *Proceedings of the 12th Workshop on Challenged Networks*. 2017, pp. 63–68.

[164] Nils Richerzhagen, Björn Richerzhagen, Michael Walter, Dominik Stingl, and Ralf Steinmetz. "Buddies, not enemies: Fairness and performance in cellular offloading." In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2016, pp. 1–9.

[165] D Robins. "Complex event processing." In: *Second International Workshop on Education Technology and Computer Science. Wuhan*. Citeseer. 2010, pp. 1–10.

[166] Umair Sadiq, Mohan Kumar, Andrea Passarella, and Marco Conti. "Modeling and simulation of service composition in opportunistic networks." In: *ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. 2011, pp. 159–168.

[167] Umair Sadiq, Mohan Kumar, Andrea Passarella, and Marco Conti. "Service composition in opportunistic networks: A load and mobility aware solution." In: *IEEE Transactions on Computers* 64.8 (2015), pp. 2308–2322.

[168] Ozgur Sanli, Ibrahim Korpeoglu, and Adnan Yazici. "Rule-based Inference and Decomposition for Distributed In-network Processing in Wireless Sensor Networks." In: *Knowledge and Information Systems* (2016), pp. 1–34.

[169] Paolo Santi. "Topology control in wireless ad hoc and sensor networks." In: *ACM computing surveys (CSUR)* 37.2 (2005), pp. 164–194.

[170] Mahadev Satyanarayanan. "Mobile Computing: the next decade." In: *ACM workshop on mobile cloud computing & services: social networks and beyond*. 2010.

[171] Gregor Schiele, Christian Becker, and Kurt Rothermel. "Energy-efficient cluster-based service discovery for ubiquitous computing." In: *ACM SIGOPS European workshop*. 2004, p. 14.

[172] Immanuel Schweizer, Roman Bärtl, Benedikt Schmidt, Fabian Kaup, and Max Mühlhäuser. "Kraken.me mobile: the energy footprint of mobile tracking." In: *IEEE Mobile Computing, Applications and Services (MobiCASE)*. 2014, pp. 82–89.

[173] Keith L Scott and Scott Burleigh. "Bundle protocol specification." In: *RFC*. 2007.

[174] Samarth H Shah and Klara Nahrstedt. "Predictive location-based QoS routing in mobile ad hoc networks." In: *IEEE International Conference on Communications (ICC)*. Vol. 2. 2002, pp. 1022–1027.

[175] Jian Shen, Sangman Moh, and Ilyong Chung. "Routing protocols in delay tolerant networks: A comparative survey." In: *International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. 2008, pp. 6–9.

[176] Manolis Sifalakis, Basil Kohler, Christopher Scherb, and Christian Tschudin. "An information centric network for computing the distribution of computations." In: *Conference on Information-Centric Networking*. ACM. 2014, pp. 137–146.

[177] Bernard Sklar. "Rayleigh fading channels in mobile digital communication systems. I. Characterization." In: *IEEE Communications magazine* 35.9 (1997), pp. 136–146.

[178] Vasco NGJ Soares, Joel JPC Rodrigues, and Farid Farahmand. "GeoSpray: A geographic Routing Protocol for Vehicular Delay-tolerant Networks." In: *Information Fusion* 15 (2014), pp. 102–113.

[179] John Soldatos, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean-Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, et al. "Openiot: Open source internet-of-things in the cloud." In: *Interoperability and open-source solutions for the internet of things*. Springer, 2015, pp. 13–25.

[180] Zheng Song, Chi Harold Liu, Jie Wu, Jian Ma, and Wendong Wang. "Qoi-aware multitask-oriented dynamic participant selection with budget constraints." In: *IEEE Transactions on Vehicular Technology* 63.9 (2014), pp. 4618–4632.

[181] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." In: *ACM SIGCOMM workshop on Delay-tolerant networking*. 2005, pp. 252–259.

[182] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility." In: *IEEE Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2007, pp. 79–85.

[183] Utkarsh Srivastava, Kamesh Munagala, and Jennifer Widom. "Operator placement for in-network stream query processing." In: *ACM symposium on principles of database systems*. 2005, pp. 250–258.

[184] Fabrice Starks and Thomas Peter Plagemann. "Operator placement for efficient distributed complex event processing in manets." In: *IEEE Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2015, pp. 83–90.

[185] Ralf Steinmetz and Lars Wolf. "Quality of Service: Where are We?" In: *IFIP Workshop on Quality of Service (IWQOS)*. 1997, pp. 210–221.

[186] Artur Sterz, Lars Baumgärtner, Ragnar Mogk, Mira Mezini, and Bernd Freisleben. "DTN-RPC: Remote procedure calls for disruption-tolerant networking." In: *IFIP Networking*. 2017, pp. 1–9.

[187] Sagar A Tamhane, Mohan Kumar, Andrea Passarella, and Marco Conti. "Service composition in opportunistic networks." In: *IEEE Green Computing and Communications (GreenCom)*. 2012, pp. 285–292.

[188] Sunil Taneja and Ashwani Kush. "A survey of routing protocols in mobile ad hoc networks." In: *International Journal of innovation, Management and technology* 1.3 (2010), p. 279.

[189] Cristiano Tapparello, Colin Funai, Shurouq Hijazi, Abner Aquino, Bora Karaoglu, He Ba, Jiye Shi, and Wendi Heinzelman. "Volunteer Computing on Mobile Devices: State of the Art and Future Research Directions." In: *Mobile Computing and Wireless Networks: Concepts, Methodologies, Tools, and Applications*. 2016, pp. 2171–2198.

[190] Thiago Teixeira and Michael Zink. "Evaluating Information-Centric Networks in Disconnected, Intermittent, and Low-Bandwidth Environments." In: *ACM Symposium on Architectures for Networking and Communications Systems*. 2016, pp. 135–136.

[191] Julian Timpner and Lars Wolf. "Query-response geocast for vehicular crowd sensing." In: *Ad Hoc Networks* 36 (2016), pp. 435–449.

[192] Julian Timpner, Mario Wozenilek, and Lars Wolf. "Breadcrumb routing: Query-response geocast for mobile originators in vehicular networks." In: *IEEE Vehicular Networking Conference (VNC)*. 2014, pp. 45–52.

[193] Christian Tschudin and Manolis Sifalakis. "Named functions and cached computations." In: *IEEE Consumer Communications and Networking Conference (CCNC)*. 2014, pp. 851–857.

[194] Guliz S Tuncay, Giacomo Benincasa, and Ahmed Helmy. "Participant recruitment and data collection framework for opportunistic sensing: a comparative analysis." In: *ACM MobiCom workshop on challenged networks*. 2013, pp. 25–30.

[195] Güliz Seray Tuncay, Giacomo Benincasa, and Ahmed Helmy. “Autonomous and distributed Recruitment and Data Collection Framework for Opportunistic Sensing.” In: *ACM SIGMOBILE Mobile Computing and Communications Review* 16.4 (2013), pp. 50–53.

[196] Gareth Tyson, John Bigham, and Eliane Bodanese. “Towards an information-centric delay-tolerant network.” In: *IEEE Computer Communications Workshops (INFOCOM workshops)*. 2013, pp. 387–392.

[197] Gareth Tyson, Nishanth Sastry, Ruben Cuevas, Ivica Rimac, and Andreas Mauthe. “A survey of mobility in information-centric networks.” In: *ACM Communications* 56.12 (2013), pp. 90–98.

[198] Gareth Tyson, Nishanth Sastry, Ivica Rimac, Ruben Cuevas, and Andreas Mauthe. “A survey of mobility in information-centric networks: challenges and research directions.” In: *ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*. 2012, pp. 1–6.

[199] Gabriel Urzaiz, David Villa, Felix Villanueva, and Juan Carlos Lopez. “Process-in-Network: a comprehensive network processing approach.” In: *Sensors* 12.6 (2012), pp. 8112–8134.

[200] Amin Vahdat and David Becker. *Epidemic Routing for partially connected Ad Hoc Networks*. Tech. rep. Duke University, 2000.

[201] Andras Varga and Rudolf Hornig. “An overview of the OMNeT++ simulation environment.” In: *International conference on Simulation tools and techniques for communications, networks and systems & workshops*. 2008, p. 60.

[202] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah M’hamed. “Sparse mobile crowdsensing: challenges and opportunities.” In: *IEEE Communications Magazine* 54.7 (2016), pp. 161–167.

[203] Lucas Wang, Alexander Afanasyev, Romain Kuntz, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. “Rapid traffic information dissemination using named data.” In: *ACM workshop on emerging name-oriented mobile networking design-architecture, algorithms, and applications*. 2012, pp. 7–12.

[204] Qi Wang and John E Taylor. “Quantifying human mobility perturbation and resilience in Hurricane Sandy.” In: *PLoS one* 9.11 (2014).

[205] Thomas Watteyne, Antonella Molinaro, Maria Grazia Richichi, and Mischa Dohler. “From manet to ietf roll standardization: A paradigm shift in wsn routing protocols.” In: *IEEE Communications Surveys & Tutorials* 13.4 (2011), pp. 688–707.

[206] Haoyi Xiong, Daqing Zhang, Guanling Chen, Leye Wang, and Vincent Gauthier. “Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint.” In: *IEEE Pervasive Computing and Communications (PerCom)*. 2015, pp. 55–62.

[207] Dianlei Xu, Yong Li, Xinlei Chen, Jianbo Li, Pan Hui, Sheng Chen, and Jon Crowcroft. "A Survey of Opportunistic Offloading." In: *IEEE Communications Surveys & Tutorials* (2018).

[208] Yong Yao, Johannes Gehrke, et al. "Query Processing in Sensor Networks." In: *Cidr*. 2003, pp. 233–244.

[209] Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. "A socio-aware overlay for publish/subscribe communication in delay tolerant networks." In: *ACM symposium on modeling, analysis, and simulation of wireless and mobile systems*. 2007, pp. 225–234.

[210] Yu-Ting Yu, Raheleh B Dilmaghani, Seraphin Calo, MY Sanadidi, and Mario Gerla. "Interest propagation in named data manets." In: *IEEE Computing, Networking and Communications (ICNC)*. 2013, pp. 1118–1122.

[211] Xiangshen Yu, Rodolfo WL Coutinho, Azzedine Boukerche, and Antonio AF Loureirol. "A distance-based interest forwarding protocol for vehicular information-centric networks." In: *IEEE Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–5.

[212] Weiwei Yuan, Donghai Guan, Eui-Nam Huh, and Sungyoung Lee. "Harness human sensor networks for situational awareness in disaster reliefs: a survey." In: *IETE Technical Review* 30.3 (2013), pp. 240–247.

[213] Arkady Zaslavsky, Prem Prakash Jayaraman, and Shonali Krishnaswamy. "Sharelikescrowd: Mobile analytics for participatory sensing and crowd-sourcing applications." In: *IEEE Data Engineering Workshops (ICDEW)*. 2013, pp. 128–135.

[214] Deze Zeng, Song Guo, Ivan Stojmenovic, and Shui Yu. "Stochastic modeling and analysis of opportunistic computing in intermittent mobile cloud." In: *IEEE Industrial Electronics and Applications (ICIEA)*. 2013, pp. 1902–1907.

[215] Daqing Zhang, Leye Wang, Haoyi Xiong, and Bin Guo. "4W1H in Mobile Crowd Sensing." In: *IEEE Communications Magazine* 52.8 (2014), pp. 42–48.

[216] Ke Zhang, Yuming Mao, Supeng Leng, Quanxin Zhao, Longjiang Li, Xin Peng, Li Pan, Sabita Maharjan, and Yan Zhang. "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks." In: *IEEE Access* 4 (2016), pp. 5896–5907.

[217] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. "Named data networking." In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73.

[218] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. "Incentives for mobile crowd sensing: A survey." In: *IEEE Communications Surveys & Tutorials* 18.1 (2016), pp. 54–67.

[219] Yang Zhang, Dusit Niyato, and Ping Wang. "Offloading in mobile cloudlet systems with intermittent connectivity." In: *IEEE Transactions on Mobile Computing* 14.12 (2015), pp. 2516–2529.

[220] Yanru Zhang, Lingyang Song, Walid Saad, Zaher Dawy, and Zhu Han. "Contract-based incentive mechanisms for device-to-device communications in cellular networks." In: *IEEE Journal on Selected Areas in Communications* 33 (2015), pp. 2144–2155.

[221] Dong Zhao, Huadong Ma, and Liang Liu. "Energy-efficient opportunistic coverage for people-centric urban sensing." In: *Wireless networks* 20.6 (2014), pp. 1461–1476.

[222] Dong Zhao, Huadong Ma, Shaojie Tang, and Xiang-Yang Li. "COUPON: A cooperative framework for building sensing maps in mobile opportunistic networks." In: *IEEE transactions on parallel and distributed systems* 26.2 (2015), pp. 392–402.

[223] Xuejun Zhuo, Wei Gao, Guohong Cao, and Yiqi Dai. "Win-Coupon: An incentive framework for 3G traffic offloading." In: *IEEE International Conference on Network Protocols (ICNP)*. 2011, pp. 206–215.

[224] Jakub Zkebala, Piotr Cikepka, and Adam RezA. "Pedestrian acceleration and speeds." In: *Probl. Forensic Sci.* 91 (2012), pp. 227–234.

All web pages cited in this work have been checked in July 2018. However, due to the dynamic nature of the World Wide Web, their long-term availability cannot be guaranteed.

A

APPENDIX

A.1 TESTBED EVALUATION WITH ADAPTIVE TASK-ORIENTED MESSAGE TEMPLATE

We implemented the adaptive task-oriented message template—ATMT concept as introduced in Chapter 4 using Java, to demonstrate that distributed processing concept with ATMT can be utilized for generic applications [3–5]. Java is a cross-platform programming language, allowing us to port the implementation to run on Raspberry Pis and on Android-based smartphones, which are the intended devices for the testbed. Overall, our developed testbed comprises a mix of hardware devices, and a testbed controller. The testbed controller does not take part in the processing of ATMT; its only purpose is to setup and manage the deployment of the devices. When a participating device is started, it can choose a role, registers itself, and maintains a socket connection with the testbed controller to wait for further commands. Four roles are implemented in the testbed: sensor nodes, which provide sensing data, delegator nodes which have the domain knowledge of how to process the sensing data and constructs the complete ATMT message with operations graph, operator nodes with one or more available services to take part in the processing, and end nodes that expect to receive the end result of the task execution. Using the interface of the testbed controller, a direct Transmission Control Protocol (TCP) connection can be set up among devices by drawing a line between two nodes. Thereby, the testbed controller allows us to flexibly set up arbitrary topologies for test-cases. Further features serving the purpose of simplifying the deployment include, for example, saving a created topology as preset for reuse or resetting links of all nodes. A test case for ATMT processing is triggered as soon as the sensor node sends data to its neighbors; accordingly, the ATMT will be further forwarded for the operator nodes to process, until it reaches the end node. Each node logs for itself the statistics desired for evaluation, such as the number of received ATMT packets and the number of sent/processed ATMT packets, network traffic.

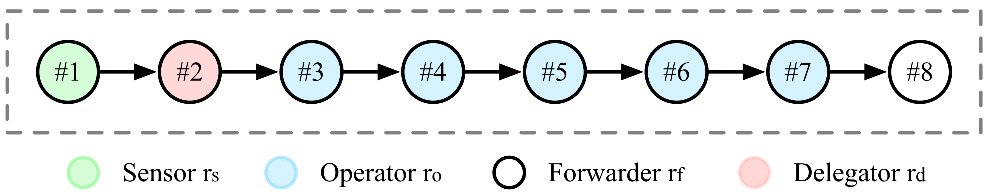


Figure 38: Raspberry Pis in the testbed are connected to form a daisy chain topology.

We used 8 Raspberry Pi devices for the testbed, which are connected in a daisy chain topology as shown in Figure 38. Several ATMT messages are sent through this topology. Each node uses the CPU load as an indicator for overload to handover

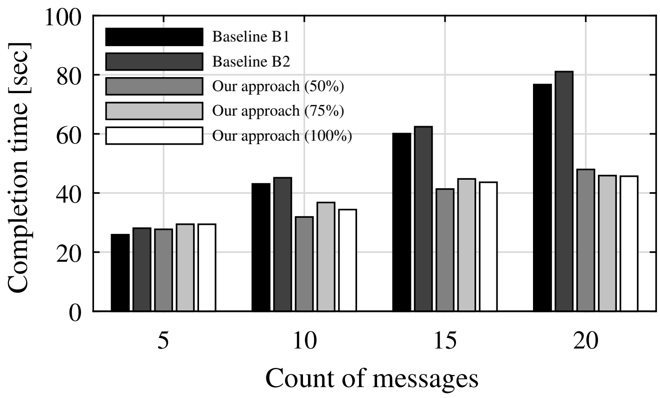


Figure 39: Completion time of ATMT tasks measured with the testbed

upcoming ATMT messages. We compare the distributed processing approach with varying thresholds for CPU (50%, 75%, 100%), against two baselines—(B1) processing all data at the first node and forwarding the total results, and (B2) forwarding the whole data and let the last node process all. The results shown in Figure 39 confirm that a distributed processing using ATMT concept can reduce the completion time, with the higher number of tasks, and that if devices are willing to share more computing resource, the overall performance can be improved, since the processing tasks will be distributed among devices.

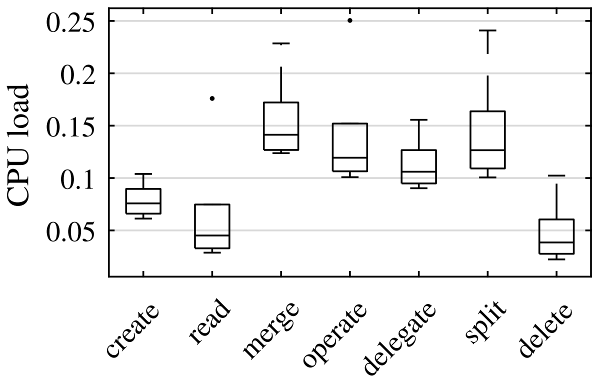


Figure 40: CPU measured of graph-based primitives with a Raspberry Pi during test

In Chapter 4, we have defined seven graph-based primitives, which enable the modification of ATMT messages directly in the network. These are *create, read, merge, operate, delegate, split, and delete*. In the evaluation with the testbed, we measured the CPU load, caused by each graph-based primitive individually on a Raspberry Pi device. The results are shown in Figure 40. We can observe, that the CPU load for all operations on the device utilize less than 0.25%. Given that Raspberry Pi device is considered as a low-power commodity hardware, the measured values are nearly negligible in modern mobile devices, such as smart phones, tablets.

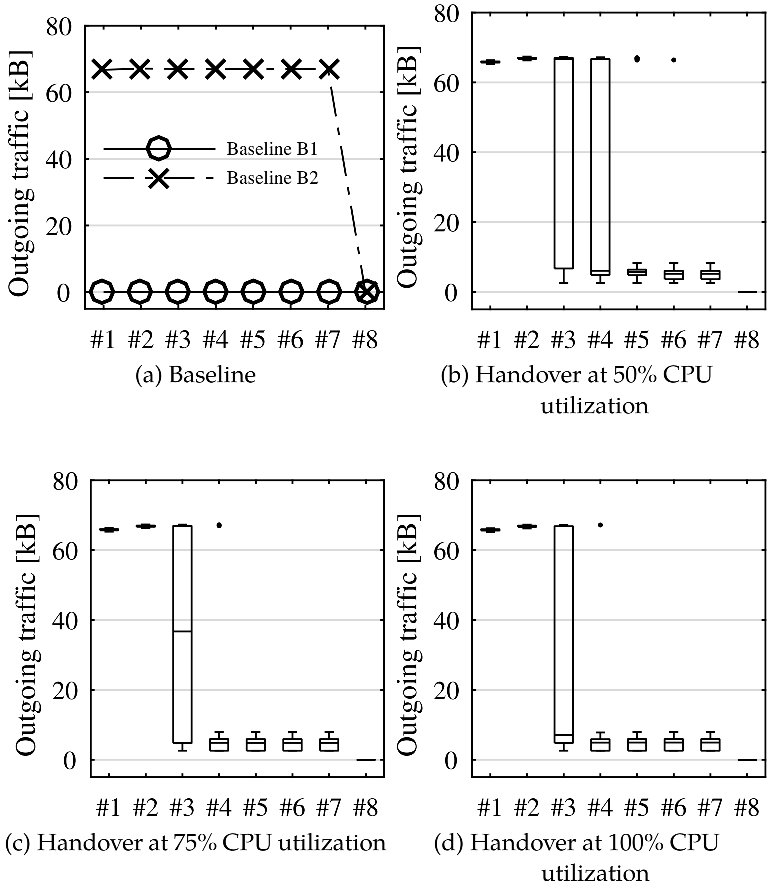


Figure 41: Network traffic measurement when executing ATMT tasks

We also evaluated the network traffic flowing through the participating devices in the testbed to assess study the effect of distributed in-network processing. We sent 20 ATMT messages in burst through the topology shown in Figure 38. Each message contains a processing task, that required the devices to perform a clustering algorithm on attached payload data. The network traffic is measured as outgoing traffic at each participating device. The results are shown in Figure 41. Again, we used the two baselines B1, B2, and the CPU utilization (50%, 75%, 100%) as metric for local strategies to handover ATMT tasks. Baseline B1 generates very low network traffic since all tasks are performed in the beginning and only the results will be forwarded, Thus B1 can serve as the lower bound. Baseline B2 generates the highest network traffic, since the whole unprocessed payload data will be forwarded through the network. Thus B2 can serve as the upper bound. In contrast, our ATMT concept allows for self-organizing distributed processing, in which, each device can perform part of the processing task thus the processing is distributed among participating devices. If each device offers up to 50% of its CPU utilization, the network traffic can be decreased at device #5 indicating that the first three operators execute most of the tasks. With 75%, and 100%, the first device can take over most of the processing tasks, thus the network traffic can be decreased already starting at device #4. Overall, the evaluation results confirm the need for designing local handover strategies to balance between computation and network traffic among devices.

A.2 LIST OF ACRONYMS

AoI	Area Of Interest
API	Application Programming Interface
CEP	Complex Event Processing
CPU	Central Processing Unit
CS	Content Store
DAG	Directed Acylic Graph
DIFS	Distributed Inter-Frame Space
DTN	Delay-tolerant Networking
FIB	Forwarding Information Base
ICN	Information-centric Networking
IoT	Internet Of Things
IRI	Internationalized Resource Identifiers
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
NDN	Named Data Networking
NFN	Named Function Networking
PIT	Pending Interest Table
QoI	Quality Of Information
QoS	Quality Of Service
REST	Representational State Transfer
RPC	Remote Procedure Call
RSU	Road Size Units
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UUID	Universally Unique Identifier
V2V	Vehicle-to-vehicle
VANET	Vehicular Ad Hoc Network
WSN	Wireless Sensor Network

A.3 SUPERVISED STUDENT THESES

[1] Pratyush Agnihotri. “Towards Information centric Collaboration Framework for Mobile Crowd Sensing.” Master Thesis. TU Darmstadt, 2015.

[2] Simon Farr. “Middleware zur robusten Kommunikation von freiwilligen Feuerwehren im Einsatzfall.” Master Thesis. TU Darmstadt, 2014.

[3] Hasan Hazem. “QoS aware opportunistic Services Composition for In-Network Processing.” Master Thesis. TU Darmstadt, 2017.

[4] Mathias Hornjak. “Analyse und Einsatz von IoT-Technologien im Parkraummanagement.” Master Thesis. TU Darmstadt, 2018.

[5] Prasanna Mahadevaswamy. “Development of efficient Recruitment Strategies for Participatory Sensing.” Master Thesis. TU Darmstadt, 2015.

[6] Sooraj Mandotti. “Robust Information Delivery in Hybrid Mesh Networks through Breadcrumb based Connectivity Forcasting.” Master Thesis. TU Darmstadt, 2017.

[7] Marius Rettberg-Päplow. “Verfahren zur fairen Lastverteilung für komplexe In-Netzwerk Datenverarbeitung.” Master Thesis. TU Darmstadt, 2017.

[8] Ali Haider Rizvi. “Situation-Aware Complex Event Processing over Information-Centric.” Master Thesis. TU Darmstadt, 2018.

[9] Peter Schmidt. “Kollaborative Middleware für Mobile Crowd Sensing.” Bachelor Thesis. TU Darmstadt, 2015.

[10] Dominik Schneider. “Ein QoS model basierter Ansatz zur Vernetzung des Industrial Internet of Things.” Master Thesis. TU Darmstadt, 2017.

[11] Thulasiram Valleru. “Efficient Operator Migration for Distributed Complex Event Processing in Dynamic User Environments.” Master Thesis. TU Darmstadt, 2015.

B

AUTHOR'S PUBLICATIONS

MAIN PUBLICATIONS

[1] The An Binh Nguyen, Pratyush Agnihotri, Christian Meurisch, Manisha Luthra, Rahul Dwarakanath, Jeremias Blendin, Doreen Böhnstedt, Michael Zink, and Ralf Steinmetz. "Efficient Crowd Sensing Task Distribution Through Context-aware NDN-based Geocast." In: *42nd Conference on Local Computer Networks (LCN)*. Oct. 2017, pp. 52–60.

[2] The An Binh Nguyen, Frank Englert, Simon Farr, Christian Gottron, Doreen Böhnstedt, and Ralf Steinmetz. "Hybrid Communication Architecture for Emergency Response - An Implementation in Firefighter's Use Case." In: *Proceedings of the fifth International Workshop on Pervasive Networks for Emergency Management (PerNEM)*. Mar. 2015, pp. 528–533.

[3] The An Binh Nguyen, Christian Klos, Christian Meurisch, Patrick Lampe, Björn Richerzhagen, and Ralf Steinmetz. "Demo: Enabling In-Network Processing utilizing Nearby Device-to-Device Communication." In: *IFIP Networking*. May 2018, A7–A8.

[4] The An Binh Nguyen, Christian Meurisch, Stefan Niemczyk, Doreen Böhnstedt, Kurt Geihs, Max Mühlhäuser, and Ralf Steinmetz. "Adaptive Task-Oriented Message Template for In-Network Processing." In: *Proc. International Conference on Networked Systems (NetSys)*. Mar. 2017, pp. 1–8.

[5] The An Binh Nguyen, Christian Meurisch, Stefan Niemczyk, Christian Klos, Doreen Böhnstedt, and Ralf Steinmetz. "Demo: Facilitating Volunteer Computing Resources for In-Network Processing through Message Template." In: *International Conference on Networked Systems (NetSys)*. Mar. 2017, pp. 1–2.

[6] The An Binh Nguyen, Marius Rettberg-Päplow, Christian Meurisch, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. "Complex Services Offloading in Opportunistic Networks." In: *IFIP Networking*. May 2018, pp. 532–540.

[7] The An Binh Nguyen, Melanie Siebenhaar, Ronny Hans, and Ralf Steinmetz. "Role-based Templates for Cloud Monitoring." In: *IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC)*. Dec. 2014, pp. 242–250.

CO-AUTHORED PUBLICATIONS

[8] Alaa Alhamoud, Pei Xu, Frank Englert, Philipp Scholl, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "Evaluation of User Feedback in Smart Home for Situational Context Identification." In: *Proceedings of the 2015 IEEE*

International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). Mar. 2015, pp. 20–25.

[9] Rahul Dwarakanath, Boris Koldehofe, Yashas Bharadwaj, The An Binh Nguyen, David Eyers, and Ralf Steinmetz. "TrustCEP: Adopting a Trust-Based Approach for Distributed Complex Event Processing." In: *IEEE International Conference on Mobile Data Management (MDM)*. 2017, pp. 30–39.

[10] Frank Englert, Marius Rettberg-Päplow, Sebastian Kössler, Alaa Alhamoud, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "Enhancing User Privacy by Data Driven Selection Mechanisms for Finding Transmission-Relevant Data Samples in Energy Recommender Systems." In: Mar. 2015.

[11] Ronny Hans, Melanie Holloway, The An Binh Nguyen, Alexander Müller, Marco Seliger, and Jürgen Lang. *Cloud-Applikationen in der Finanzindustrie - Dienstgütemerkmale und deren Überwachung*. Tech. rep. Sept. 2017.

[12] Florian Jomrich, Markus Grau, Tobias Meuser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "ICCOMQS - Intelligent measuring framework to ensure reliable communication for highly automated vehicles." In: *Proceedings of 2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, Nov. 2017, pp. 311–318.

[13] Patrick Lieser, Nils Richerzhagen, Tim Feuerbach, The An Binh Nguyen, Björn Richerzhagen, Doreen Böhnstedt, and Ralf Steinmetz. "12entralized Resource Allocation in Mobile Networks." In: *Demonstrations of the 42nd IEEE Conference on Local Computer Networks (LCN Demonstrations)*. Oct. 2017, pp. 1–3.

[14] Christian Meurisch, Julien Gedeon, Artur Gogel, The An Binh Nguyen, Fabian Kaup, Florian Kohnhäuser, Lars Baumgärtner, Milan Schmittner, and Max Mühlhäuser. "Temporal Coverage Analysis of Router-based Cloudlets Using Human Mobility Patterns." In: *GLOBECOM*. Dec. 2017.

[15] Christian Meurisch, Julien Gedeon, The An Binh Nguyen, Fabian Kaup, and Max Mühlhäuser. "Decision Support for Computational Offloading by Probing Unknown Services." In: *International Conference on Computer Communications and Networks (ICCCN)*. July 2017.

[16] Christian Meurisch, The An Binh Nguyen, Julien Gedeon, Florian Kohnhäuser, Milan Schmittner, Stefan Niemczyk, Stefan Wullkotte, and Max Mühlhäuser. "Upgrading Wireless Home Routers as Emergency Cloudlet and Secure DTN Communication Bridge." In: *International Conference on Computer Communications and Networks (ICCCN): Poster.* July 2017.

[17] Christian Meurisch, The An Binh Nguyen, Stefan Wullkotte, Stefan Niemczyk, Florian Kohnhäuser, and Max Mühlhäuser. "NICER911: Ad-hoc Communication and Emergency Services Using Networking S03tphones and Wireless Home Routers." In: *International Symposium on Mobile Ad Hoc Netwroking and Computing (MobiHoc'17): Posters*. July 2017.

[18] Tobias Meuser, Patrick Lieser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "Adaptive Information Aggregation for Application-specific Demands." In: *Proceedings of the 1st International Balkan Conference on Communications and Networking*. May 2017.

[19] Tobias Meuser, Patrick Lieser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. "Lifetime-Aware Information Aggregation under Uncertainty for Advanced Driver Assistance Systems." In: *Ad Hoc Networks* (2018), pp. 56–67.

[20] Tobias Meuser, Björn Richerzhagen, Ioannis Stavrakakis, The An Binh Nguyen, and Ralf Steinmetz. "Relevance-Aware Information Dissemination in Vehicular Networks." In: *Proceedings of IEEE WoWMoM*. 2018.

[21] Melanie Siebenhaar, The An Binh Nguyen, Ulrich Lampe, Dieter Schuller, and Ralf Steinmetz. "Concurrent Negotiations in Cloud-based Systems." In: *Proceedings of the 8th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON)*. Springer-Verlag Berlin Heidelberg, Feb. 2012, pp. 17–31.

CURRICULUM VITÆ

PERSONAL

Name	The An Binh Nguyen
Date of Birth	March 24, 1987
Place of Birth	Hanoi, Vietnam
Nationality	Vietnam

EDUCATION

Since 04/2014	Technische Universität Darmstadt, Doctoral candidate at the department of Electrical Engineering and Information Technology
04/2011 - 12/2013	Technische Universität Darmstadt, Master studies in Computer Science, Degree: Master of Science
04/2008 - 03/2011	Technische Universität Darmstadt, Bachelor studies in Computer Science, Degree: Bachelor of Science
09/2007 - 03/2008	Technische Universität Darmstadt, Studienkolleg

ACADEMIC EXPERIENCE

Since 04/2014	Researcher funded by Hessen-LOEWE-funded project – NICER "Networked Infrastructureless Cooperation for Emergency Reponse"
Since 04/2014	Researcher associated by BMBF-funded project – SMARTER "Smartphone-based Communication Networks for Emergency Response"

WORK EXPERIENCE

Since 04/2014	Technische Universität Darmstadt, Research assistant at the researcher group

Distributed Sensing Systems at the
Multimedia Communications Lab (KOM)

TEACHING ACTIVITY

2014-2018	Lecture "Communication Networks I", Teaching Assistant.
2014-2016	Seminar "Advanced Topics in Future Internet Research", Supervisor.
2014-2016	Lab "Multimedia Communications Lab/Project", Supervisor.

HONORS

03/2016 - 11/2017	Selected candidate in BMBF funded SOFTWARE CAMPUS program.

D

ERKLÄRUNGEN LAUT DER PROMOTIONSORDNUNG

§8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Datum und Unterschrift

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/`