Nils Richerzhagen, Björn Richerzhagen, Dominik Stingl, Ralf Steinmetz. *The Human Factor: A Simulation Environment for Networked Mobile Social Applications*. In Proc. International Conference on Networked Systems (NetSys), IEEE, 2017. ISBN: 978-1-5090-4394-1

The Human Factor: A Simulation Environment for Networked Mobile Social Applications

Nils Richerzhagen*, Björn Richerzhagen*, Dominik Stingl, Ralf Steinmetz
Multimedia Communications Lab
Technische Universität Darmstadt, Germany
e-mail: {firstname.lastname}@kom.tu-darmstadt.de

***Abstract*—Networked mobile social applications are becoming increasingly popular with Pokémon Go being a recent example. These applications focus on direct interaction between mobile users within close proximity. As a result, tailored communication systems have been proposed to exploit the resulting locality properties by augmenting typical cloud-based application infrastructures with local ad hoc communication. However, evaluating these communication systems is challenging: (i) client mobility heavily influences interaction and, thus, the resulting workload; (ii) a multitude of connectivity models needs to be considered for direct ad hoc communication, cellular networks, and potential Wi-Fi offloading scenarios. Consequently, we present a set of human mobility models, interaction models for networked social applications, and communication models to ease the creation of these surrounding heterogeneous scenarios for the considered communication systems. We integrate these models into a common simulation and prototyping environment, bridging the gap between mobility and network simulation and allowing the combined study of human-centric and network-centric effects. We show the applicability and resulting insights of our proposed models for two case studies: a mobile augmented reality game and a monitoring service utilizing multi-dimensional offloading.**

I. Introduction

With the increasing number of mobile devices and their extended computational capabilities, a range of networked mobile social applications emerged. Ranging from applications for communication, such as WhatsApp or Telegram, to fully fledged mobile augmented reality multiplayer games such as Google's Ingress or the recently hyped Pokémon Go, the applications focus on the direct interaction between users. For a large fraction of mobile social applications, interaction heavily depends on a user's physical location. While in some cases the location only determines the relevance of content (e.g., retrieving a list of nearby restaurants), applications such as the aforementioned mobile augmented reality games use physical proximity to determine not only the content but also the intended recipients.

This focus on local interaction with a user's vicinity motivates a plethora of communication systems and middleware solutions that utilize direct interaction between nearby users. Typical goals are (i) to lower the latency between interacting users for an increased user experience, and (ii) to reduce the load on the cellular network through content offloading or aggregation [1]. In contrast to research on pure ad hoc networks, global state and connectivity has to be maintained in parallel. Relying for instance on a cloud-based backend service, the communication between users as well with the backend service leads to a mixture of local ad hoc and cellular network connectivity. At the same time, the performance of such systems is highly dependent on the behavior of the users, especially their mobility and interaction patterns.

Consequently, when evaluating communication systems for mobile social applications, models for ad hoc and cellular communication need to be combined with application-specific mobility and social interaction models for a deep understanding of the system's performance under realistic conditions. However, according to Harri et al. [2], combining these models is a challenging and cumbersome task as mobility and network simulators are not designed to work together.

To ease the combination of simulation models from different domains and sources, we proposed the Simonstrator platform in an earlier work [3]. In this work, we present modular extensions to the Simonstrator platform that enable simulative and prototypical evaluation of networked mobile social applications. Most notably, we introduce a *mobility and social interaction model* that utilizes attraction points to model application-specific points of interest, which determine both, user mobility and interaction between users. Combined with real-world map data from OpenStreetMap and the respective navigation functionality, realistic movement profiles and interaction workloads for a wide variety of applications can be generated. By combining *connectivity models* for ad hoc communication via Wi-Fi and Bluetooth with measurement-based abstractions for the cellular network, hybrid communication systems can be studied. To address the gap identified by Rebecchi et al. [1] and to support an integrated evaluation of data offloading approaches, we add support for Wi-Fi access points and the resulting handover events within our connectivity models. We show the applicability of our simulation environment and the impact of intertwined mobility and interaction models for two case studies: a mobile augmented reality game [4] and a monitoring service [5].

The remainder of this paper is structured as follows. In Section II, we discuss mobile social applications and the characteristics of the underlying communication systems. Section III provides a brief overview of the Simonstrator platform and our extensions required to support an integrated evaluation of mobile social applications. In Sections IV and V we detail

*Authors contributed equally.

our proposed models for user mobility and social interaction and connectivity, respectively. Section VI contains case studies showing the applicability of the proposed simulation environment. Section VII discusses relevant related works and Section VIII concludes the paper.

II. Mobile Social Applications

As sketched in the introduction, our contribution fills the gap between mobility and network simulation to facilitate the accurate evaluation of networked mobile social applications. In such applications, effects of human mobility, interaction, and network connectivity are closely intertwined [6]. In the following, we detail the characteristics of our scenario and the challenges that need to be addressed during evaluation of the resulting communication systems.

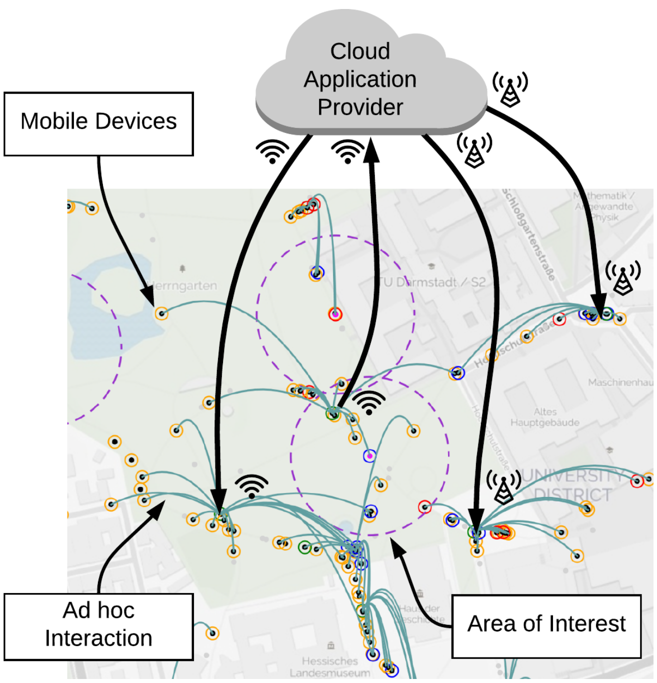


Fig. 1. Illustration of the considered scenario. Mobile users interact with nearby users within their area of interest via a cloud-based application provider, using the cellular network or Wi-Fi access points.

Figure 1 illustrates the scenario of a networked mobile social application from a communication system's perspective. Users interact with the application through their mobile devices, leading to a number of events that need to be distributed to other interested or affected users. For the scenario of a mobile augmented reality game, such events contain user-triggered interactions with in-game objects as well as (periodical) status updates. The relevance of an event for a specific player is mostly determined by its physical location, as mobile augmented reality games focus on the interaction with objects and other players in the real world. Brokering of events is usually done in the logically centralized cloud. Consequently, the generated data of a user needs to be uploaded via the cellular network or – if available – a local Wi-Fi network.

As the availability and communication characteristics of a cellular or Wi-Fi connection strongly depend on the physical location and, thus, are subject to user mobility, the performance of the communication system directly depends on user movement. During movement, handovers between cellular and Wi-Fi connectivity may occur, requiring corresponding connectivity models (c.f. Section V). User movement, in turn, also depends on information available within the mobile application: in the case of a mobile augmented reality game, players are more likely to move towards specific points of interest that are tied to in-game features. At the same time, the social aspects of the application lead to the formation of groups of players that interact for a longer period of time. Often, this is supported through application-specific structures, such as guilds or teams a player belongs to. This motivates the design of our modular mobility and social interaction model (c.f. Section IV).

Due to the mostly local interaction between users, several proposed communication systems try to utilize local ad hoc connectivity to distribute relevant events to nearby users. However, these systems still rely on the logically centralized, cloud-based broker for coordination and maintenance of the global state and for other purposes such as accounting and cheat detection. Consequently, models for device discovery and direct ad hoc communication – e.g. via Bluetooth or Wi-Fi ad hoc – need to complement the cellular connectivity models within the evaluation framework. In the following, we will discuss the Simonstrator environment and our additions that allow the evaluation of the developed communication systems for networked mobile social applications.

III. Extending the Simonstrator

This section provides a brief overview of the Simonstrator platform as introduced in [3] and the integration of the extensions proposed in this paper. The Simonstrator is a component-based simulation and prototyping platform targeted towards the evaluation of networked mobile applications and services written in Java. Its goal is to enable flexible combinations of different models and data sources by wrapping all access to such sources via common APIs. It is a perfect fit to bridge the gap between network, interaction, and mobility models in a scenario where they all influence each other.

Within the Simonstrator, each user is modeled as a *host*, that is configured with a set of *components*. These components realize specific functional aspects of a device: network and transport-layer functionality for connectivity, fully-fledged services and overlays on the application layer as well as access to sensors for e.g., the location or the current battery level, is provided. While the framework only contains the interface definitions, the actual implementations can be plugged in from a number of runtime environments or external libraries, as described in detail in [3]. Applications that are to be evaluated can also provide implementations of specific components. This framework allows to mix and match different types of models and environments depending on the intended scenario and goal of the evaluation. One key design aspect is the support for prototypical deployments on Android devices, which aides in the collection of real measurement data. For more details on the platform the interested reader is referred to [3].

An overview of the proposed simulation environment, highlighting our extensions to the original Simonstrator platform, is provided in Figure 2. Core of the proposed simulation environment is the Simonstrator framework consisting of a set of APIs and utility functions. We added a graph-based abstraction for social ties between users. This serves as a data source for the movement models proposed in this paper as well as the interaction model of the respective application. Furthermore, we added APIs to manipulate movement from within an application's interaction model (c.f. Section IV). The movement model has an impact on the connectivity characteristics, as the user's position is used to retrieve connection parameters within the connectivity models (c.f. Section V).

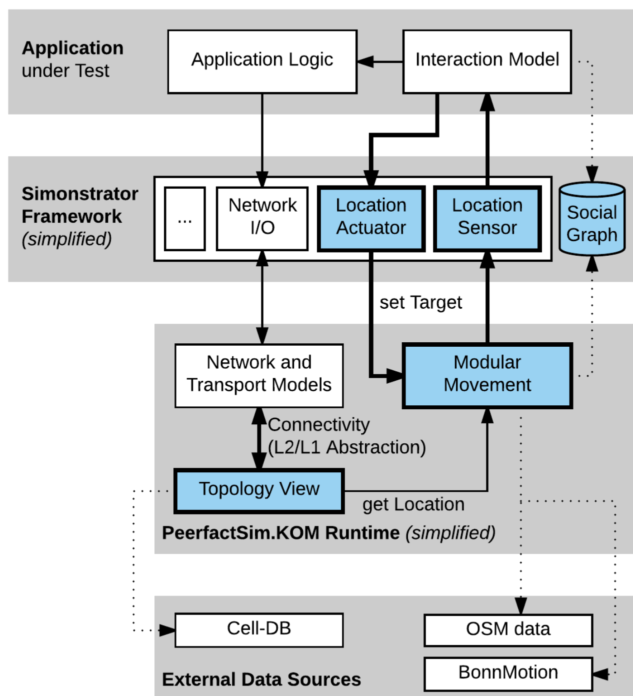


Fig. 2. Platform overview with highlighted contributions. The interaction model of the application can interact with mobility models and thereby also with connectivity models. Social ties among users are modeled as a graph and are used to determine interactions as well as mobility characteristics.

The models themselves are implemented within the PeerfactSim.KOM runtime environment [7], extending our earlier works on mobility models [8]. To support offloading scenarios, we further integrate APIs for the handover between Wi-Fi access points and the cellular network, as well as support for direct ad hoc connectivity. We provide corresponding connectivity models that realize relevant properties of the scenario for the PeerfactSim.KOM runtime environment. It is important to note that for the scenario of mobile social applications we are not interested in, for example, physical layer accuracy of the LTE transmission model. Instead, we are interested in higher-layer effects, such as the performance of offloading via ad hoc connections to neighboring users or via available Wi-Fi access points. Therefore, the proposed models rely on insights from popular measurement studies and further support the simple integration of real measurement data, as discussed in the following.

IV. Mobility and Social Interaction

As previously discussed, node movement affects connectivity and the workload of the respective application. Additionally, as a novelty for mobile social applications, movement in itself is affected by the application and its users. On the one hand, the application uses real-world locations as points of interest and thereby incentivizes users to approach and interact with these locations. On the other hand, social applications foster direct interaction with friends and nearby users, leading to social connections between users of the application. Consider a group of friends as an example: if one user within the group is attracted by an application-specific point of interest, he will most probably communicate that to his friends – not necessarily by application-provided means – and thereby influence the movement of the whole group.

To enable applications – or workload generators – to interact with the respective movement models, we extend the LocationSensor API provided by the Simonstrator [3]. Via the LocationSensor, applications can request the current location of a user and register for periodic updates, much like the corresponding Android API. We add a LocationActuator API to allow manipulation of the target location that is approached by a node. To support the respective operation in a generic fashion, we propose our Modular Movement Model, as detailed in the following.

A. The Modular Movement Model

To cater for the peculiarities of the scenario, we designed a layered movement model that consists of four basic steps with the respective strategies, as illustrated in Figure 3. The model starts by generating and placing a set of *attraction points*. Depending on the respective strategy, attraction points can be generated based on OpenStreetMap data or by providing traces. For the scenario of mobile social applications, and especially for the aforementioned augmented reality games, attraction points are usually given by the application and correspond to application-specific points of interest. Next, each mobile node is assigned to one of the attraction points. Again, different *assignment strategies* are supported and further discussed in this section: a simple random assignment, an assignment strategy that mimics properties of the well-known SLAW movement model [9], and an assignment based on social ties (c.f. Section IV-D).

Nodes move towards their assigned attraction point, following the rules of the configured node movement model. A simple model is to use direct linear movement (as proposed in SLAW), however, more realistic map-based movement is usually utilized in our scenario. Once a node reaches the

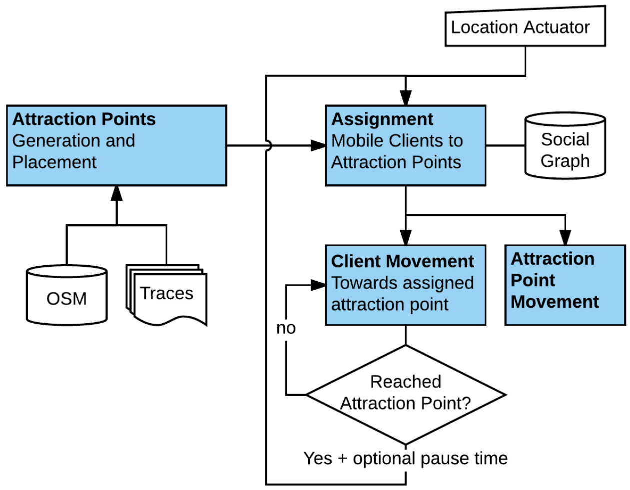


Fig. 3. Flowchart of the Modular Movement Model, composed out of four basic strategies. Movement of nodes and attraction points can utilize the same models. By incorporating social ties between users during the assignment process, realistic mobility and interaction models can be generated.

respective attraction point, it may or may not pause for a given time before triggering a re-assignment to a new attraction point. The aforementioned location actuator API allows triggering of this assignment procedure at any point in time. It further supports the creation of new attraction points and enables fixed assignment of the respective nodes to any desired attraction point. In contrast to other proposals in the literature, attraction points do not need to be fixed to a given location. Similar to mobile nodes, they can be configured with a movement model and float around.

This leads to some interesting properties with respect to the mobility models that can be realized using our approach. For example, to simulate basic Random Waypoint mobility, one just needs to assign the Random Waypoint mobility model to attraction points, create as many attraction points as mobile users and assign each attraction point to exactly one user. While this might sound complex for such a simple model, it also allows to configure arbitrary group-based mobility models. However, the model becomes even more powerful with the integration of the BonnMotion mobility trace generator and OpenStreetMap for routing and the creation of attraction points, as discussed in the following.

B. Integrating BonnMotion Movement Traces

As BonnMotion [10] already includes output generators and parsers for a number of formats, it is rather simple to utilize BonnMotion-generated movement traces for node and/or attraction point movement. All BonnMotion models generate full movement paths for each node. These traces can directly be used for the creation of attraction points and their assignment: each BonnMotion node is transformed into an AttractionPoint, and then assigned to a simulated mobile node. In the end, attraction points move around according to the BonnMotion trace, and each attraction point is directly followed by a single node. For the whole duration of the simulation, nodes are not re-assigned to other attraction points. Thereby, one can easily compare the BonnMotion movement profiles against movement patterns that originate from other configurations of our model.

C. OpenStreetMap Routing and Attraction Points

While the model is able to mimic less complex movement models, its core strength results from the utilization of real-world map data when combined with direct interaction via the location actuator API. By combining OpenStreetMap data of the desired area with the GraphHopper[a] offline routing library, reproducible experiments with realistic node movement patterns are achieved. By relying on the routing library, nodes move from one attraction point to the other following pedestrian walkways or streets. As the application can freely determine the next attraction point via the location actuator API, sequences of meaningful application-specific places can be visited. This becomes especially interesting for the case of mobile augmented reality games, where crawlers usually only report the location of in-game interaction (i.e., the attraction points) and not the route that was taken in-between those points. Combining such a trace of visited attraction points with the powerful OpenStreetMap routing capabilities leads to highly realistic scenarios.

As mentioned, for a number of scenarios the set of attraction points is already specified by the application. However, in other use cases, attraction points might simply correspond to locations such as public parks or specific amenities. As already detailed in our previous work [8], our toolchain supports the automatic extraction of attraction points from OpenStreetMap data in such cases. However, to fully mimic the behavior of users in mobile social applications, the *social* aspect of mobility has to be considered as well for the flexible assingment of attraction points.

D. Modeling Social Ties between Users

Social relationships between users are an important factor in mobile social applications and networks [11]. They lead to potentially increased contact times between specific users and consequently impact the efficiency of offloading approaches relying on direct ad hoc communication. Therefore, as illustrated in Figure 3, the assignment of nodes to their attraction points can be configured to rely on a graph-based abstraction for social ties, the so-called *social graph*. Relying on graphs to represent social ties is a well-known practice, as discussed in Section VII. Within the modular movement model, the social graph can be used to realize patterns where (i) groups of friends tend to meet at the same attraction point, (ii) specific sets of attraction points are more frequently visited by a specific set of users (e.g., students are attracted by the *university*), (iii) a single node's action influences the movement of a group of users (in augmented reality games, this is often observed as part of a reaction to in-game actions).

[a]https://github.com/graphhopper/graphhopper, accessed 09/12/2016

Within the social movement model groups of friends are represented by clusters of nodes in the social graph. The mobility model can be configured such that it selects the same attraction points for nodes within such clusters at a higher probability, thereby leading to group-like mobility patterns. At the same time, the social graph can also be utilized within the application or workload, thereby determining not only network parameters but also interaction characteristics. Mobile social applications and networking mechanisms relying on device-to-device communication can be examined with great detail using the social movement model graph properties.

As briefly discussed before, altering the social graph during runtime, for example as a consequence of a user interaction within the application, also affects the resulting mobility characteristics. In scenarios where social ties might not be known or available via traces, graph structures can be loaded from GraphML[b] files or generated via the JUNG-generator[c]. For convenience, generators for Kleinberg Small World [12] and Eppstein Power Law [13] graphs are already included. Both, imported and generated graphs are stored for future repeatability of experiments.

The Modular Movement Model provides a simple yet powerful abstraction for a wide range of mobility models. Coupled with OSM data and a model for social ties between users, it is a powerful tool to mimic real-world behavior of users interacting with a mobile social application. We evaluate selected properties of the model in Section VI. By allowing manipulation of the movement model through the LocationActuator API as well as through the social graph abstraction, the desired coupling between interaction and mobility is achieved. In the following section we detail the implications on the connectivity model as well as our abstraction for heterogeneous networking scenarios.

V. hetGrid Connectivity Model

Besides the impact of social interaction and movement, mobile social applications are characterized by the heterogeneity of network connectivity and available communication interfaces. Access to the Internet is enabled via public or private Wi-Fi access points (APs) or 5G/LTE networks. Direct local communication between nearby devices can be realized via Wi-Fi ad hoc or Bluetooth. Especially for offloading scenarios, multiple communication interfaces and means for a handover in between these interfaces need to be provided by an evaluation environment.

The connectivity models within the Simonstrator realize physical and data link layer functionality. In contrast to most existing network simulators (c.f. Section VII), we focus on measurement and trace-based abstractions for lower-layer effects rather than integrating complex models. An overview of the connectivity models utilized in our scenario is provided in Figure 4. We extend the Simonstrator with the hetGrid connectivity model that is specifically targeted towards offloading use-cases. It supports cellular connectivity and connections to Wi-Fi access points, as discussed in the following. The model is complemented with a handover sensor that notifies applications of changes in the network connectivity – i.e., a handover from Wi-Fi to cellular networks and vice versa. For direct ad hoc communication, the corresponding models are already provided by the Simonstrator [3].

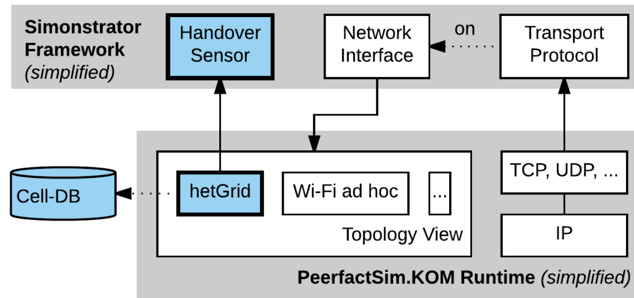


Fig. 4. Targeted towards offloading scenarios, a novel hetGrid connectivity model is proposed that can easily integrate measurement databases. Applications are notified via the Handover Sensor API, if connectivity changes.

To leverage the resources of large-scale measurement databases for 5G/LTE networks [14] and urban Wi-Fi AP coverage [15], the hetGrid model divides the simulation area into freely configurable cells. For simplicity, rectangular cells are considered in the following – however, often used hexagonal cells can also be modeled. Instead of modeling physical properties, we rely on the results available through the aforementioned databases to assign a quality function $q_c(n)$ to each cell c. The function depends on the number of currently active users n within the cell and returns a tuple $(latency, bandwidth, drop)$ of connection parameters that are valid for the current connection. As measurement databases usually only report average values without knowledge about the number of users, a simple implementation of q_c would just return the static values for the respective area from the measurement database.

However, as shown by Shafiq et al. [16], the performance characteristics of a cell depend on the number of users in that cell. Therefore, more complex functions can be used to model overload situations and the corresponding quality degradations, as illustrated in Figure 5. By assigning different functions to cells, scenarios with certain weak spots in terms of communication can be simulated. This is especially interesting for disaster response scenarios, where the availability and reliability of communication infrastructure can be limited around certain locations.

The hetGrid model contains a second layer of cells that corresponds to public or private Wi-Fi access points. Usually, these cells are smaller and do not cover the whole simulation area, leading to spots without Wi-Fi coverage. Access to Wi-Fi cells is assigned to individual users or groups of users upon configuration. For this purpose, information from the social graph can also be used (for example, to enable access to private

[b]http://graphml.graphdrawing.org/, accessed 08/26/2016

[c]http://jung.sourceforge.net/index.html, accessed 08/26/2016

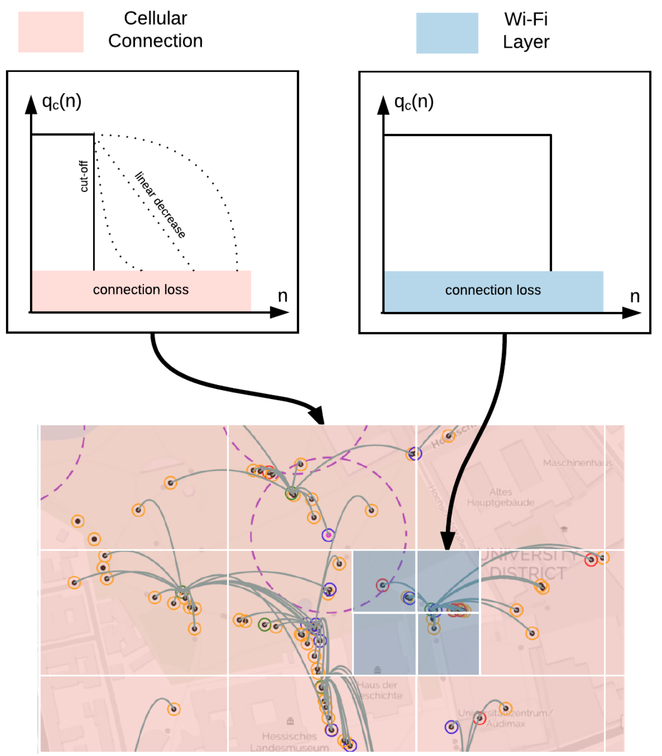


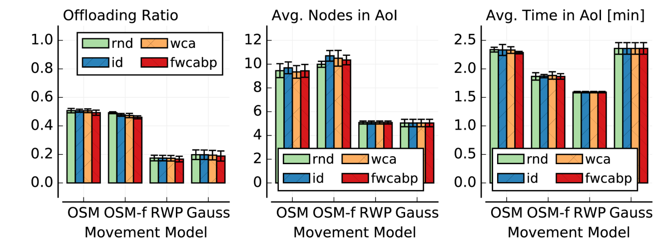
Fig. 5. The hetGrid model with per-cell performance functions as illustrated. The model supports client handover between Wi-Fi and cellular connections.

infrastructure to friends). Whenever a user establishes or loses a connection to a Wi-Fi access point, the handover sensor is triggered. This allows an application to react for example by offloading traffic of other nearby users [1].

VI. Case Studies

Within this section, we show the applicability of our proposed models as well as insights obtained in two different case studies. The simulation environment and our proposed models were used during the evaluation of two distinct adaptive services for networked mobile social applications: (i) a modular publish/subscribe middleware [4] for location-based applications, and (ii) a transition-enabled monitoring service [5], [17]. Both services focus on offloading through the utilization of access points and direct ad hoc connectivity. Thus, understanding the impact of user mobility on the performance of the service is crucial for the design of efficient offloading algorithms.

Within the modular publish/subscribe middleware [4], locally relevant events are distributed by selected gateway nodes utilizing direct ad hoc connectivity. Consequently, redundant transmissions via the cellular network connection are reduced. The achieved offloading ratio (i.e., the fraction of traffic that is no longer sent via the cellular connection) of the system is shown in Figure 6(a) for four scenarios that only differ in the utilized movement model. The choice of the gateway selection algorithm (rnd, id, wca, fwcabp) only has a minor impact on the system performance. However, whether



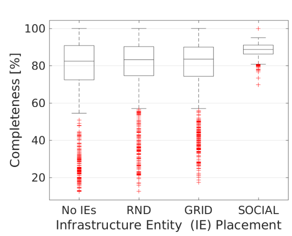
(a) Offloading Ratio (b) Nodes in AoI (c) Time in AoI

Fig. 6. Offloading performance and scenario characteristics for different mobility models. OSM and OSM-f are based on pedestrian mobility, while RWP and Gauss are synthetic models used as a baseline. Taken from [4].

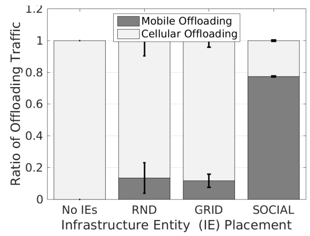
the system is evaluated using OpenStreetMap-based mobility within our modular movement model (OSM and OSM-f with reduced pause times) or using simpler models such as Random Waypoint or Gaussian Movement has significant impact.

This is due to the fact that for mobile social application, location and mobility also determines the application interaction. In Figures 6(b) and 6(c) two application-specific characteristics are shown: the number of other nodes within a specific node's area of interest (Fig. 6(b)) and the average time a node spends in another node's area of interest (Fig. 6(c)). Unsurprisingly, attraction points used in the OSM-based models lead to denser clusters of nodes around such attraction points. Consequently, the average number of nodes is higher than for RWP and Gauss, leading to higher efficiency of the proposed gateway approach as content is relevant to a larger audience. Considering only the time a node spends within the area of interest of another node, the Gaussian mobility model achieves similar results as the OSM-based model in its default configuration, whereas Random Waypoint leads to shorter contact times. During the design of more sophisticated offloading strategies, a more realistic mobility model is crucial. Otherwise, the respective strategies will be tailored towards characteristics that cannot be observed under real-world conditions, leading to a potentially significant loss in performance. It is important to note that the location of attraction points as well as pause times used in the evaluation were derived from real-world user traces [4].

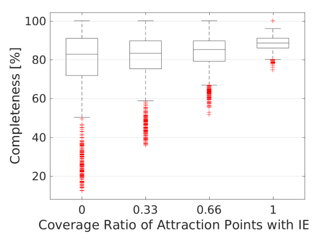
In evaluations of the transition-enabled monitoring service [5], [17] the importance of heterogeneous network coverage via access points and cellular network connectivity has been observed. Relying on multiple communication means in the offloading process of monitoring data the handover between these technologies as well as a varying coverage of the former is a key impact factor for the achieved quality of the monitoring. The modular social movement model, with which the representation of varying social interactions between users has been made possible, the monitoring service has been evaluated in urban scenarios with different placements and coverages of access points. The results clearly indicate the importance of (i) a movement model capable to represent group formations due to social relations and (ii) a network



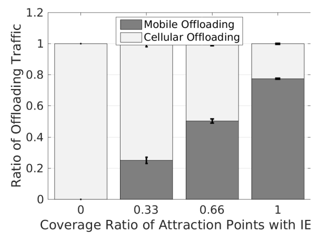
(a) Impact of the placement of access points on the performance of the monitoring service.



(b) Impact of the placement of access points on the offloading ratio of the monitoring service.



(c) Impact of the coverage of access points on the performance of the monitoring service.



(d) Impact of the coverage of access points on the offloading ratio of the monitoring service.

Fig. 7. Impact of the placement and coverage of access points on a monitoring service relying on offloading via multiple communication means. Evaluated in a scenario with social interactions between users, plot taken from [5].

model able to provide for varying network characteristics. Especially the placement and coverage of access points and changing characteristics of the cellular network in form of connectivity parameters have a great impact.

From Figure 7(a) and 7(c) reveal that the both the placement of access points and the coverage ratio of social interesting places with access points are important impact factors for the achieved quality of the monitoring service. The achieved quality is measured using the completeness metric. The completeness of the monitoring service describes the percentage of how many of the potentially possible monitoring information is collected from the nodes in a given interval. The results show the importance of understanding node mobility and social interaction in mobile social networks. Beside the performance of the system, the offloading performed by the system is highly depending on the placement as well as the coverage of social relevant places achieved by access points as visible in Figure 7(b) and 7(d).

VII. Related Work

Several simulation frameworks and environments have been proposed and are actively maintained by the research community. In the following, we distinguish between (i) network-centric and (ii) mobility-centric approaches and study the applicability of the respective approach to the scenario of networked mobile social applications. We further discuss alternatives for modeling social ties and interaction, as this constitutes an important aspect of our proposed models.

With respect to network-centric simulations, ns-2 and its successor ns-3 [18] are probably the most prominent examples. Both offer a framework for event-based simulations and come with a number of models for connectivity, covering cellular networks (GSM, LTE), ad hoc networks (including car-to-car). Support for node mobility is enabled through an abstraction layer, and some basic reference models are provided. Other data sources, such as BonnMotion [10], can be included – however, interaction is unidirectional. Recently, support for map-based movement through the use of the Google Maps API has been proposed [19]. However, by depending on an online navigation service with potentially non-deterministic results (e.g., re-routing due to traffic conditions), the repeatability and also the scalability of simulations is limited. Furthermore, coupling is only uni-directional, thereby limiting the applicability in our scenario. By relying on an offline navigation library and OSM data, experiments conducted in our environment are reproducible and larger-scale scenarios with thousands of nodes can be simulated.

The Veins project [20] also aims at combining network simulation with accurate movement models in a bidirectional fashion. Veins explicitly targets vehicular network simulations and integrates the network models of OMNeT++/MiXiM [21] with the vehicular mobility simulator SUMO. While our environment can be adapted to vehicular scenarios, the focus of both approaches differs significantly. Veins focuses more on lower-layer effects of car-to-car communication under realistic road conditions, while our environment is targeted towards mobile devices and their heterogeneous communication means when interacting with cloud-based services. Veins and our proposed environment share the design concept of bi-directional interaction between movement and network-related models for more realistic evaluation scenarios.

As a well-established generator for mobility traces, BonnMotion [10] provides a large number of models, ranging from simple synthetic to more complex map-based models. Recently, area-based models for more interactive disaster response scenarios have been added [22]. However, due to the nature of BonnMotion being a generator, it cannot directly react to network simulations and the respective workload, thereby limiting the applicability of the models to our scenario. However, as stated previously, support for BonnMotion-generated movement traces is also included in our models as a baseline to assess the impact of movement being influenced by the application itself.

In [23], the authors combine the well known SLAW model [9] for attraction points and their selection strategies with map-based local movement models. While SLAW is used to determine an attraction point for a client, the client moves towards the attraction point following real-world streets and walkways. Similar to our findings, the authors state that the map-based movement has a significant impact for the use case of opportunistic networking. This further underlines the importance of sophisticated mobility models when evaluating systems that heavily depend on client mobility.

Musolesi et al. [24] utilized a social graph within their mobility model. The authors assigned random interaction probabilities to the edges of the graph. Nodes with stronger interaction probabilities will tend to group during movement.

Movement can be influenced by changing the interaction probabilities over time. We extend this concept by using the social graph as a gateway between an application's interaction model and the modular movement model.

Harri et al. [2] state in their survey of mobility models for vehicular ad hoc networks that mobility and network simulators are not designed to work together. We believe that we are able to bridge this gap by using the Simonstrator platform and, thus, enabling the integration of a plethora of data sources and models via a unified framework. Our proposed extensions to the platform explicitly address the dependencies in-between movement, interaction, and connectivity to allow for a more realistic evaluation of mobile social applications and related communication aspects.

VIII. Conclusion

In this work we proposed a simulation environment specifically designed for the challenges of evaluating networked mobile social applications. We motivated that user mobility, interaction, and connectivity are closely intertwined, and that current evaluation environments do not consider these dependencies. Relying on the Simonstrator platform we bridge the gap between mobility, interaction, and network simulation with a focus on the dependencies in between those worlds. We proposed a set of models that accurately reflect user behavior and relevant network characteristics while at the same time enabling easy integration of traces and measurements. We detail the implications of our models for selected case studies, highlighting the importance of application-specific models to get a better understanding on the benefits of offloading approaches and local ad hoc communication in heterogeneous and complex scenarios.

Our simulation environment can further be altered or extended[d] to support a wide range of location-based communication scenarios not considered in this paper. One example are emergency response scenarios after large-scale disasters, where movement models are strongly determined by the application logic and the social interaction of the respective entities. Another use case are automotive scenarios, where information retrieved via an application or communication system results in a behavior change of the driver (e.g., a route update), again requiring combined mobility and interaction models.

Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of projects B1, C2 and C3 within the Collaborative Research Centre (CRC) 1053 – MAKI.

References

[1] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data Offloading Techniques in Cellular Networks: A Survey," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 2, pp. 580–603, 2015.

[2] J. Härri, F. Filali, and C. Bonnet, "Mobility Models for Vehicular ad hoc Networks: A Survey and Taxonomy," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 4, pp. 19–41, 2009.

[3] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications," in *Proc. SIMUTOOLS*. ICST, 2015, pp. 99–108.

[4] B. Richerzhagen, N. Richerzhagen, S. Schönherr, R. Hark, and R. Steinmetz, "Stateless Gateways – Reducing Cellular Traffic for Event Distribution in Mobile Social Applications," in *Proc. Conference on Computer Communication and Networks (ICCCN)*. IEEE, 8 2016, pp. 1–9.

[5] N. Richerzhagen, B. Richerzhagen, R. Hark, D. Stingl, and R. Steinmetz, "Limiting the Footprint of Monitoring in Dynamic Scenarios through Multi-dimensional Offloading," in *Proc. Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2016, pp. 1–9.

[6] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, "Applications, Architectures, and Protocol Design issues for Mobile Social Networks: A Survey," *Proceedings of the IEEE*, vol. 99, no. 12, 2011.

[7] D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz, "PeerfactSim.KOM: A Simulation Framework for Peer-to-peer Systems," in *Proc. High Performance Computing and Simulation (HPCS)*. IEEE, 2011, pp. 577–584.

[8] D. Stingl, B. Richerzhagen, F. Zöllner, C. Gross, and R. Steinmetz, "PeerfactSim.KOM: Take it back to the Streets," in *Proc. High Performance Computing and Simulation (HPCS)*. IEEE, 2013, pp. 80–86.

[9] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A new Mobility Model for Human Walks," in *Proc. INFOCOM*. IEEE, 2009, pp. 855–863.

[10] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A Mobility Scenario Generation and Analysis Tool," in *Proc. SIMUTOOLS*, 2010, pp. 51:1–51:10.

[11] X. Hu, T. H. Chu, V. C. Leung *et al.*, "A Survey on Mobile Social Networks: Applications, Platforms, System Architectures, and Future Research Directions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1557–1581, 2015.

[12] J. Kleinberg, "The Small-world Phenomenon: An Algorithmic Perspective," in *ACM STOC*, 2000, pp. 163–170.

[13] D. Eppstein and J. Y. Wang, "A Steady State Model for Graph Power Laws," in *WWW WebDyn*, 2002.

[14] F. Kaup, F. Michelinakis, N. Bui, J. Widmer, K. Wac, and D. Hausheer, "Behind the NAT - A Measurement based Evaluation of Cellular Service Quality," in *Proc. CNSM*, 2015, pp. 228–236.

[15] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile Data Offloading: How Much Can WiFi Deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, 2013.

[16] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "A First Look at Cellular Network Performance During Crowded Events," in *ACM SIGMETRICS*, 2013.

[17] N. Richerzhagen, B. Richerzhagen, M. Walter, D. Stingl, and R. Steinmetz, "Buddies, not Enemies: Fairness and Performance in Cellular Offloading," in *Proc. WoWMoM*. IEEE, 6 2016, pp. 1–9.

[18] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.

[19] T. Cerqueira and M. Albano, "RoutesMobilityModel: Easy Realistic Mobility Simulation using external Information Services," CISTER Research Center, Polytechnic Institute of Porto (ISEP-IPP), Tech. Rep. CISTER-TR-150302, 5 2015.

[20] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[21] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ the MiXiM Vision," in *Proc. Simulation Tools and Techniques (SIMUTOOLS)*. ICST, 2008, pp. 71:1–71:8.

[22] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, "Modeling Mobility in Disaster Area Scenarios," *Performance Evaluation*, vol. 66, no. 12, pp. 773 – 790, 2009.

[23] M. Schwamborn and N. Aschenbruck, "Introducing Geographic Restrictions to the SLAW Human Mobility Model," in *International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, 2013, pp. 264–272.

[24] M. Musolesi and C. Mascolo, "Designing Mobility Models based on Social Network Theory," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 3, pp. 59–70, 2007.

[d] Git repository at `www.simonstrator.com`. Please contact the authors for full access, as some code is not yet publicly available.