An Experimental Frame for Evaluating Service Trading Overlays in Mobile ad-hoc Networks

Mathias Röhl, Adelinde M. Uhrmacher
University of Rostock, Institute of Computer Science
mroehl,lin@informatik.uni-rostock.de

Birgitta König-Ries
Friedrich-Schiller-Universitt Jena, Institute of Computer Science
koenig@informatik.uni-jena.de

Abstract: For evaluating dynamics of mobile ad-hoc networks at least three different layers have to be distinguished: the application layer, the protocol layer, and the network layer. We present an experimental frame that has been developed in the simulation system James II to support experimentation with mobile information systems, as developed in DIANE (*Dienste in Ad-hoc Netzen, services in ad-hoc networks*). The flexibility of the approach is demonstrated by plugging different user models and protocol implementations into the experimental frame. The experimental frame is used to explore network load for the different settings.

1 Introduction

Mobile information systems are often based on the agent metaphor, interpreting a mobile information system as a dynamic community, each of which members offers and requests information services. Generally, developing multi-agent systems is considered to be of an intrinsically experimental and exploratory nature. Not surprisingly, simulation is widely used to test strategies in mobile information systems [AY06]. In a simulation-based testing of strategies, these models become explicit and the subject of experimentation [GHRU06]. A suitable model of the environment at an adequate level of abstraction is obviously central in achieving valid results [WPM^{+}05]. In simulating mobile information systems (or adhoc networks in general) three different layers are distinguished: the application layer, the protocol layer, and the network layer. If the protocols are the subject of testing and evaluation, the environmental model refers to network and application layer. The level of detail varies in capturing the essentials of the application layer. Users' behaviors are described by combining mobility models with service usage and provision models. Both are often based on stochastic, independent distributions [JM96, LH99]. This approach has a major drawback: If we want to evaluate methods developed for the application layer, realistic models of user behavior are needed. For instance, how well a service trading approach works depends heavily on the distribution of the nodes in the network, their density/sparsity in certain areas, their relative movement patterns, their usage models. If the simulation is not fed with realistic data here, we cannot expect to obtain valid evaluation results. Lately, more elaborate users' models have been introduced. E.g., daily activity schedules [KRKB04] attempt to mimic the access and provision of services more realistically. Depending on the system under study, and the questions to be answered, the level of detail of the environmental model, i.e. the network model and the users' model, varies. To support an experimentation with varying levels of detail, flexible and modular modeling and simulation frameworks are required.

In this paper we will present an experimental frame that has been developed in the simulation system JAMES II [Mod06, HU07] to support experimentation with mobile information systems, as developed in DIANE. The flexibility of the approach will be demonstrated by different user and protocol models that are plugged into the experimental frame. The experiments described in this paper evaluate methods developed within the DIANE project using the JAMES II modeling and simulation framework. In the following section, we give a brief overview of the aims of the DIANE project.

2 Services in Ad-Hoc Networks

The aim of DIANE (*Dienste in Ad-hoc Netzen, services in ad-hoc networks*) is the efficient and effective provisioning of services in ad-hoc networks. The motivation here is that each mobile device in an ad-hoc network possesses rather limited capabilities. In order to provide all the functionality and information that is needed from the user's point of view, it therefore becomes necessary to use external resources. This should be handled as transparently as possible for the user, i.e. she should not be required to manually search for appropriate resources and manually bind them to her applications. Instead, this should be done automatically by a middleware running on her device. Service-orientation is a suitable paradigm to achieve this resource sharing. However, some extensions to standard realizations of service-oriented architectures are needed to achieve the desired transparency in dynamic, open environments such as ad-hoc networks. Within DIANE, we have developed solutions for a number of these issues: There is a service offer and request description language that can be used together with matchmaking and composition algorithms to automatically find and bind services, there is a distributed reputation system to ensure fair cooperation. Additionally, a mechanism is needed to ensure that service offers and requests can find each other. In traditional settings, this is typically achieved via a repository. In ad-hoc networks such a centralized repository is not viable, since it cannot be guaranteed that any node one will be constantly available. In the literature, distributed hash tables (DHT) have been proposed as an alternative. While adaptations of DHTs that are specifically tailored to the rather dynamic environment of ad-hoc networks work rather well, they do not allow for a sophisticated semantic search, but are basically limited to keyword search. Such a rough matching is not sufficient in our scenario. We therefore propose *Lanes* [KKRO03], a semi-semantic overlay for service trading. In this overlay, nodes are divided into groups (lanes) based roughly on physical proximity at join time. A node knows the neighboring nodes within its own lane and the anycast addresses of neighboring lanes. Within each lane, information about all the services offered by lane members is replicated on each node, i.e., each node knows about all the services offered within its lane. When a node is looking for a service, it first checks whether that service is available within its own lane. If this is not the case, the node sends the request via anycast to the neighboring lanes. There, it is processed and further distributed if necessary. Overall, the Lanes protocol consists at least of the following subprotocols. Additionally protocols can be added for optimization purposes.

- Login: When a new node wants to join the lanes structure, it sends a broadcast message. This message is answered by all nodes within one hop distance. The node chooses one of the answering nodes as its predecessor in the lane and enters the lane between this node and the next. This decision is based on physical proximity.
- Lane Maintenance: To ensure that the lanes structure remains intact, nodes exchange ping messages with their neighbors in the lane. If these messages fail, the lane broken protocol is started.
- Lane Broken: This protocol contains repair mechanisms for the lane.
- Service Search: When a node looks for a service and cannot find it within the own lane, it sends an anycast message to the neighboring lanes. The receiving node there checks whether the requested service is available in its lane and forwards the message to its neighboring lane. If a service is found, an answer is sent to the requesting node.
- Logoff: If a node decides to leave the lanes structure it should inform its neighbors. These are then able to bridge the gap without much effort. If a node leaves the message and fails to inform its neighbors, they will eventually notice this and initiate the lanes broken protocol.

3 Setup of the Experimental Frame

Simulation is an experiment performed on a model and “a model M of a system S and an experiment E is anything to which E can be applied in order to answer questions about S” [Min65]. This definition of model emphasizes that a model is not developed for a system “per se”, but always for a combination of a system to be analyzed and questions to be asked. According to this definition, multiple objectives require multiple models. The concept of *experimental frame* has been introduced to model experimental assumptions and system’s requirements explicitly [Zei84].

While a basic evaluation of the lanes protocol has been done earlier [KHMM04], we want to gain insights into the following questions:

- How does the performance compare between lanes and flooding if higher numbers of nodes are taken into account?
- How does the performance of lanes alter, if we use different user models. Here, we compared a purely stochastic user to an activity based user and a social user, where users’ activities are partially synchronized.

- How big is the potential of lanes optimization to improve performance? As described above, lanes are formed based on physical proximity at join time. This results in initially small hop counts for intra-lane communication. In this experiment we investigate how strong the degeneration of the lanes structure is over time under different user models.

An easy composition of user and protocol models would ease the experimental evaluation of the above questions. Our modeling of mobile ad-hoc networks therefore is based on a component-oriented modeling approach [RU06]. Composite model components support the hierarchical, modular construction of models. Composite components enable the development of large models from smaller ones. Couplings connect ports of one model component to compatible ports of another component. Communication between models is only allowed along these couplings. Model structures can be specified with UML's composite structure diagrams [OMG05]. For simulating these models they are mapped automatically [R̈06] to a discrete event simulation formalism that can be executed with James II [Mod06, HU07].

Figure 1 shows the top level structure of the composite component *Manet* that incorporates the three layers of interest for simulating mobile ad-hoc networks. Each *Protocol* component is connected to a *Network* and *User* component. *User* and *Protocol* are grouped within a *Node*. For the user model to work correctly two further models have to be added to the experimental frame, namely a physical and social environment model. Thereby, all these components together, except the *Protocol* component itself, form the experimental frame for evaluating service trading protocols. The *Manet* component takes parameters to initialize the number of nodes to be simulated, the type of the user model to be used, and the type of the protocol model to be used. The two latter parameters are delegated to the *Node* component.

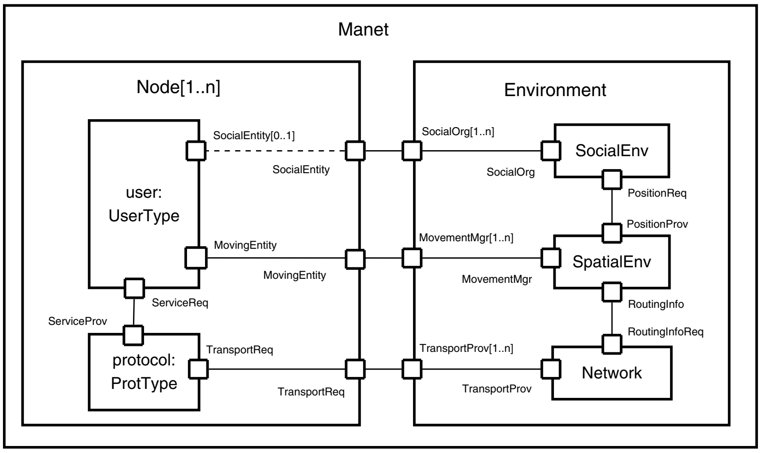

Figure 1: Overall structure of the model

The user model initiates communication with calls indicating the publication or search for services. Calls are passed to the protocol model, which answers user calls with *Response* events. This kind of interaction is expressed by the *ServiceReq* and *ServiceProv* interface respectively:

```
<interface xmlns="http://www.../javai">
  <id>ServiceReq/1.0</id>
  <eventport name="call" type="Call" isInput="false"/>
  <eventport name="response" type="Response" isInput="true"/>
</interface>

<interface xmlns="http://.../javai">
  <id>ServiceProv/1.0</id>
  <eventport name="call" type="Call" isInput="true"/>
  <eventport name="response" type="Response" isInput="false"/>
</interface>
```

The protocol component has a port which provides the *ServiceProv* interface. User components are equipped with a port declaring the presence of the *ServiceReq* interface. This is a condition for proper functioning.

For moving within the geographic environment, a user may request path information from the environment and send move events to it. Therefore, the user component has to implement the interface *MovingEntity*.

Instead of being coupled to a complementary interface at the same level the *MovingEntity* interface is delegated one level up to the node component, which exhibits the same interface. The connection between *MovingEntity* and its complement interface *MovementMgr* is done as part of the manet component.

As indicated by the dashed port and its connection to the according node port, a user may behave as a *SocialEntity*. This depends on another parameter that has to be set on the *Manet* component. So some users may participate in social interactions whereas others do not.

From the point of view of the node component, the user and protocol components are black boxes whose couplings are defined by interfaces. Each of both can be easily replaced by another that provides the same interface. We will now take a look at different implementations of user components and protocol components.

3.1 User Models

Figure 2 shows different implemenations of the user. All of these are realized themselves as composite components.

The simple user contains a *logx* subcomponent that manages login and logout behavior. If logged in, the service component becomes notified to start publishing services and searching for services randomly according to a uniform distribution. Furthermore, the *LogX* sub component selects destination points randomly and calculates routes to them. Routes are

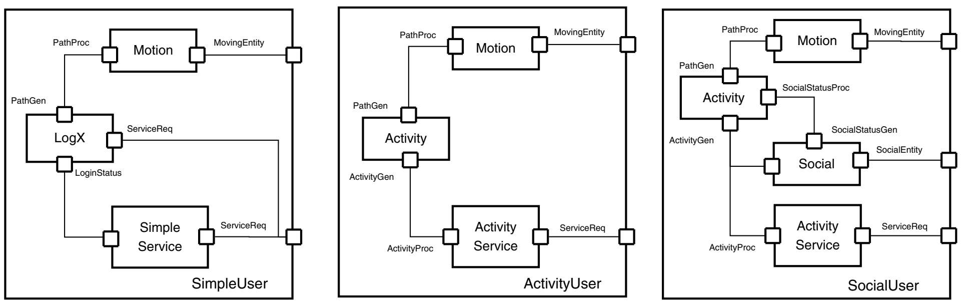

Figure 2: Three different implementations of the user component

propagated to the *Motion* sub component, which partitions the route into small steps and executed them with a certain walking speed. After reaching a destination, a new route is requested from the *LogX* component.

The second user model realizes an activity-based user behavior. Network usage and moving is not modeled independently of each other. Both depend on the activity a user is currently executing. The *Activity* model generates a schedule at the start of the day. The schedule contains fixed activities, e.g. attending a lecture as well as flexible activities, such as learning, with different priorities and durations. The current activity is passed to the *Motion* and *Service* sub component. The service component generates service offers and requests according to the received activity. Activities are not independent of the spatial context, but each location on the campus is only suited for a certain set of activities. The current activity constrains the choice of the next destination and thereby the motion model. Thus, motion and service behavior are both based on activities.

The third user model extends the activity-based model with social awareness. The sub component *Social* of each user announces planned activity to the social environment model. If users have planned similar activities and are spatially close the social environment forms a group and selects a group leader. The group leader chooses activities, which all group members may adopt. Because an activity does not uniquely define the location of performance, the group leader chooses a location out of a set of suited ones and communicates this choice to all group members. Group members are free to choose a path to the location. Thus, social users are synchronized with respect to the next joint activity and the location where this activity will be performed.

3.2 Protocols

Overlay protocols are used on top of the fourth OSI layer, i.e. the transport layer. The black-box component *Protocol* of Figure 1 can be refined similar to the user model.

A protocol takes calls from the user model and answers with response messages. In between calls and responses the protocol may contact other nodes by passing messages to the network model. Protocols communicate with the network via the *TransportReq* interface, which indicates that the protocol will send and receive events of type *Message*. These messages contain a sender address, a receiver address and the message content.

Two protocols are currently implemented, i.e. Flooding and Lanes. The wrapped message content depends on the protocol implementation. While the components representing the flooding protocol communicate with each other using *ServiceSearch* and *ServiceResponse* messages only, the Lanes component needs a number of different messages, e.g. *logIn*, *logOut*, *searchService*, *provideService*, *revokeService*.

3.3 The Environment

The environment model offers functionality with respect to the spatial, network, and social dimension. The environment consists of a map component, a network component and a social organization component.

The spatial environment represents a certain geographical area containing streets and buildings. The spatial environment keeps track of all user positions and provides users with density information of paths.

The network component models the transport layer of the network. In real applications the whole OSI stack is part of each node. Since we are interested in higher level protocols the four lower OSI layers are pooled in a centralized network component. The network component delivers messages to nodes. The connectivity of each node is calculated according to its position with respect to other nodes.

The social environment functions as a centralized communication hub with regard to social actions of the users. In order to form groups, the social environment takes the physical positions of users into account.

4 Experiments

The experimental frame introduced in the previous section is now used to evaluate protocol implementations. The basic setting is to let users represent students moving as pedestrians on a campus and perform service operations. The first series of experiments compares the lanes protocol to flooding.

4.1 Comparison of Flooding and Lanes

For the comparison the simple user model is employed. Arrival of users is uniformly distributed between 6 and 7 o clock. Users are online between 10 minutes and 60 minutes. Each user publishes 3 out of 15 different types of services after having logged into the

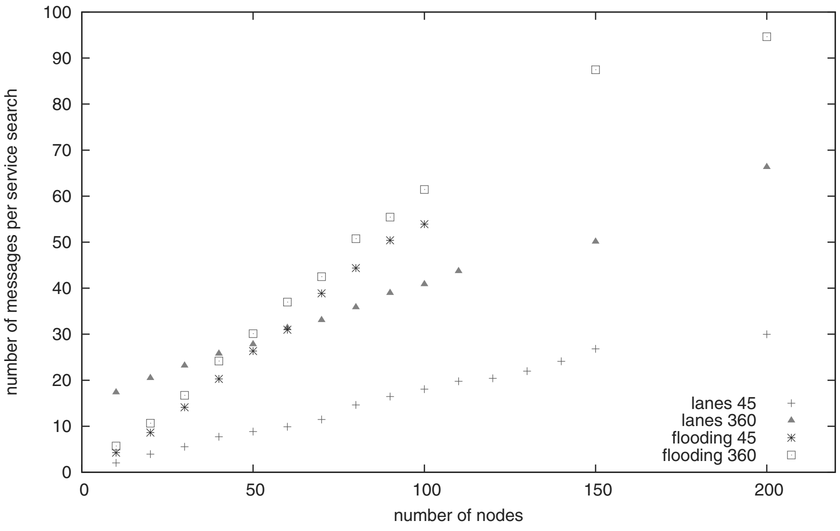

Figure 3: Messages per service search call

network. A user requests services according to a uniform distribution with expectation value 45 seconds $\pm 15s$ and $360s \pm 120s$ respectively.

Figure 3 shows the average number of messages that is required to process service searches of users. The flooding protocol does not manage an overlay structure for processing service requests. Instead it floods the whole net by iteratively propagating broadcasts. Flooding is especially well suited for highly dynamic environments with only few search requests. With increasing number of nodes broadcasting becomes expensive. For service search requests that are launched every 360 seconds on average, the break even point for managing the lanes overlay is already reached at 50 nodes. If services are requested more frequently, Lanes start to pay off already at 10 nodes.

4.2 Testing Lanes with Different User Models

Figure 4 shows three simulation runs, each with 400 nodes. Each node uses the Lanes protocol for management of service related calls. Users appear uniformly distributed between 0 and 60 minutes and immediately log into the network. Subfigure 4 a) depicts the run which uses the simple user model for generating calls. For measuring the load of the network it is distinguished between three different kinds of messages. All messages that result from building up the lane structure are summarized by the *login* trajectory. Messages for keeping the lane structure valid are subsumed under *intra lanes* messages. Messages that are used to answer service calls of users are summarized within the *inter lane* message trajectory. Using simple user models results in a quit uniformly scattered number of

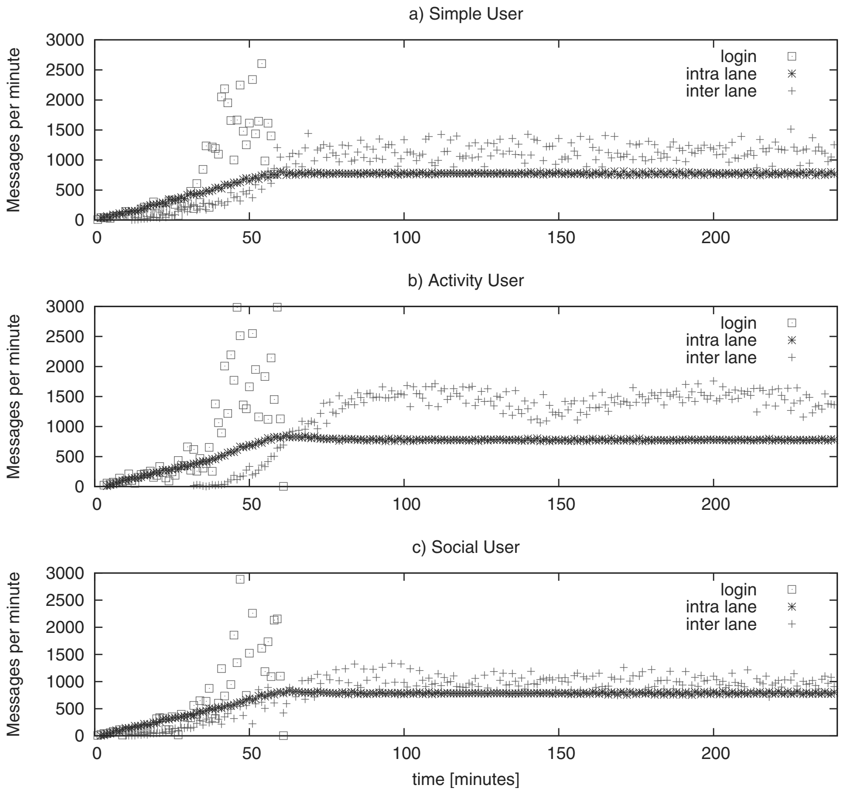

Figure 4: Absolut number of messages with 400 users

service messages during the whole online period.

The trajectories in subfigure b) are produced using the activity-based user model. The effort for service related messages varies not so much from minute to minute, but different periods with higher and lower numbers of service messages can be distinguished. The periods reflect globally scheduled activities of the users, e.g. attending a lecture.

Social user models show less distinct global behavior patterns. The different user models have, at the level of message numbers no effect on the effort for login related messages and structure managing messages.

Figure 5 shows the impact of the different user models on the effort for delivering messages, namely the average number of hops required by messages, assuming a radio range of 50 m for each node. Here, we see, that lanes indeed degenerate over time. While the rising number of hops for inter-lane communication may be due to our rather primitive implementation of the anycast mechanisms, intra-lane communication requires considerably larger hop counts after a relatively short time. This effect is particularly pronounced for the more realistic user models.

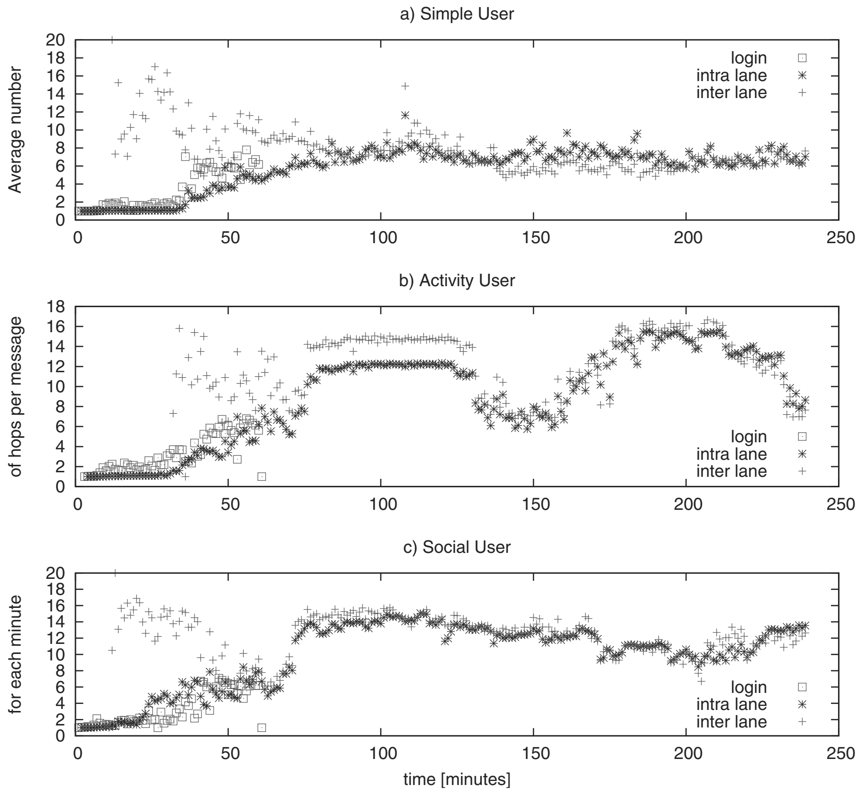

Figure 5: Average number of hops per message

5 Related Work

The main purpose of the work presented in this paper was to establish whether the James II modeling and simulation framework is suitable for evaluating approaches to service discovery in ad-hoc networks based on overlay structures. As a ”side effect” some insights into properties of the lanes overlay structure was gained. A comparison of lanes to other overlay structures including a discussion of related work in this respect can be found in [KHMM04]. Concerning the main question addressed in the article, the suitability of James II, a recent overview article [KCC05] shows the shortcomings of classical network simulators with respect to a realistic evaluation of ad-hoc networks. This paper proposes the use of a component-based modeling approach together with a general purpose modeling and simulation system (James II). Most Manet simulation studies focus on the network layer, especially on routing protocols [AY06], using simplified mobility, service usage, and radio propagation models [CBH$^+$04]. User models and their impact in evaluating mobile adhoc networks are increasingly gaining attention [KRKB04, TZH$^+$02]. The experimental frame presented in this paper is unique in its ability to comprise arbitrarily complex user models, ranging from simple users, which are based on stochastic distributions, to user models that derive mobility and network behavior from individually scheduled activities in combination with social context information. While, this paper does not directly address the realization of the network layer and the layers below, incorporation of more detailed routing protocols and radio propagation models can be composed into the experimental frame similar to the integration of different user models.

6 Conclusion

We developed an experimental frame for evaluating strategies in mobile ad-hoc networks. The presented approach allows to flexibly plug in different user models, different protocols, or different network models on demand. In this work we have focussed on inspecting the effect of different user models and protocol models. The user models varied between being purely stochastic, activity driven, and taking social interactions into account.

Based on these models we studied the performance of the lanes protocol. The experiments show an influence of the user model with respect to network loads that stem from service related calls. The user models also showed an effect on the number of hops that are required to generate and maintain lanes.

While the insights gained through these experiments are not particularly surprising, they have served as a proof of concept that our approach to modeling and simulation is indeed suitable for the class of questions related to service discovery and use in ad-hoc networks. Therefore, we intend to expand on the experiments to address further, interesting questions. Future experiments are directed towards evaluating structure-optimizing methods for the lanes protocol with dynamically adapting lanes structures. Lanes offers two of these optimization protocols: The first one aims to maintain an optimal lane length. If lanes become too short or too long, they are merged respectively split. The optimal length is not a fixed number of nodes but depends on the network (stability, load) and the ratio between service offers and requests. The second optimization protocol aims at minimizing message overhead within a lane. When the nodes drift too far apart, i.e., when hop counts for intra-lane messages increase, lanes are restructured. To ensure correct operation of these optimization protocols, transactional properties are needed.

References

[AY06] Todd R. Andel and Alec Yasinac. On the Credibility of Manet Simulations. *Computer*, 39(7):48–54, 2006.

[CBH$^+$04] Andr'es Lagar Cavilla, Gerard Baron, Thomas E. Hart, Lionel Litty, and Eyal de Lara. Simplified Simulation Models for Indoor MANET Evaluation Are Not Robust. In *Proc. SECON'04*, pages 610–620. IEEE, 2004.

[GHRU06] Martina Gierke, Jan Himmelspach, Mathias Röhl, and Adelinde M. Uhrmacher. Modeling and Simulation of Tests for Agents. In *Multi-Agent System Technologies (MATES'06)*, volume 4196/2006 of *Lecture Notes in Computer Science*, pages 49–60. Springer Berlin / Heidelberg, 2006. ISBN 978-3-540-45376-5.

[HU07] Jan Himmelspach and Adelinde M. Uhrmacher. Plug'n simulate. In *SpringSim '07*, 2007. to appear.

[JM96] David B Johnson and David A Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[KCC05] S. Kurkowski, T. Camp, and M. Colagrosso. MANET simulation studies: The Incredibles. *ACM's Mobile Computing and Communications Review*, 9(4):50–61, 2005.

[KHMM04] Michael Klein, Markus Hoffman, Daniel Matheis, and Michael Müssig. Comparison of Overlay Mechanisms for Service Trading in Ad hoc Networks. Technical Report TR 2004-2, University of Karlsruhe, October 2004.

[KKRO03] Michael Klein, Birgitta König-Ries, and Philipp Obreiter. Lanes – A Lightweight Overlay for Service Discovery in Mobile Ad Hoc Network. In *3rd Workshop on Applications and Services in Wireless Networks (ASWN2003). Berne, Swiss*, July 2003.

[KRKB04] Birgitta König-Ries, Michael Klein, and Tobias Breyer. Activity-Based User Modeling in Wireless Networks. *Mobile Networks and Applications. Special Issue on Internet Wireless Access: 802.11 and Beyond*, 2004.

[LH99] Ben Liang and Zygmunt J. Haas. Predictive Distance-Based Mobility Management for PCS Networks. In *INFOCOM (3)*, pages 1377–1384, 1999.

[Min65] Marvin Minsky. Models, Minds, Machines. In *Proc. IFIP Congress*, pages 45–49, 1965.

[Mod06] Modeling and Simulation Group, University of Rostock. James II Project description. Available online via `http://wwwmosi.informatik.uni-rostock.de/mosi/projects/cosa/james-ii` [accessed Nov 13, 2006], 2006.

[OMG05] OMG. UML Superstructure Specification Version 2.0 (Document formal/05-07-04). Available online via `http://www.omg.org/cgi-bin/doc?formal/05-07-04`, July 2005.

[Rö6] Mathias Röhl. Platform Independent Specification of Simulation Model Components. In *ECMS 2006*, pages 220–225, 2006.

[RU06] Mathias Röhl and Adelinde M. Uhrmacher. Composing Simulations from XML-Specified Model Components. In *Proceedings of the Winter Simulation Conference*, pages 1083–1090. ACM, 2006.

[TZH$^+$02] Desney S Tan, Shuheng Zhou, Jiann-Min Ho, Janak S Mehta, and Hideaki Tanabe. Design and Evaluation of an Individually Simulated Mobility Model in Wireless Ad Hoc Networks. In *Communication Networks and Distributed Systems Modeling and Simulation Conference 2002*, San Antonio, TX, 2002.

[WPM$^+$05] Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. Environments for Multiagent Systems: State-of-the-Art and Research Challenges. In *E4MAS 2004*, volume 3374 of *LNAI*, pages 1–47. Springer, 2005.

[Zei84] Bernard P. Zeigler. *Multifacetted Modelling and Discrete Event Simulation*. Academic Press, London, 1984.