The State of Simulation Tools for P2P Networks on Mobile Ad-Hoc and Opportunistic Networks

Ahmad Cheraghi, Tobias Amft, Salem Sati, Philipp Hagemeister, Kalman Graffi
Technology of Social Networks, Heinrich Heine University Düsseldorf, Germany
lastname@cs.uni-duesseldorf.de, http://tsn.hhu.de/

***Abstract*—During the Arab spring and other states of political turbulences, the access to selected servers of large communication sites and even the country-wide internet infrastructure has often been affected by enforced censorship and limited the possible communication tools. In order to overcome this censorship a full mobile decentralized network seems to be a viable option. Even though, through the strong dynamism in this decentralized mobile network, future threats, namely intermittent connectivity, changing topologies, and unreliable end nodes may occur. To handle these threats and to provide the desired functionality, self-organized protocols are necessary. This paper presents a survey on the current status of simulation tools for peer-to-peer (P2P), Opportunistic Networks (OppNet), and Mobile Ad Hoc Networks (MANET) with the aim to identify a suitable evaluation tool that is capable for combining these three networks and for simulating decentralized mobile networks. This combination will allow us to create a fully decentralized and self-organized system which cannot be blocked e.g. through the influences of political parties.**

I. Introduction

Online social networks and online real-time communication tools gained large popularity in the last decade. In the so-called Arab Spring and other states of political turbulence, these tools emerged as important for exchanging information and for organizing activities. However, maybe some became a witness of the fragility of these communication tools on the centralized internet. During these times several threats (such as blocking of access to servers of social networks (such as in Syria, Egypt, and Turkey), systematic surveillance (of US-based social networks) and even the disconnection of Egypt's internet infrastructure from the world) on communication tools were observed. To overcome these limitations and threats it is not only sufficient to aim at secure variants of server-based solutions. Moreover, a fully mobile decentralization on various layers of the communication infrastructure should be considered.

Through the advent of smartphones -as powerful computing devices with strong networking capabilities- the decentralized data storage and local (wireless) communications emerges as an option in recent year. Local wireless multi-hop communication has been a research topic for over a decade. *Mobile ad hoc networks* (MANETs) allow mobile nodes to route messages by forwarding them to nodes in their local neighborhood across the connected topology until they reach the desired destination nodes. Thus a mesh network emerges for which a wide set of routing protocols such as AODV [1], DSR [2], OLSR [3], AntSec [4] and many more have been proposed. Up to now, MANETs have not found applications which emerged in real life outside academia. However, carrying smartphones and having the option to incrementally enhance their functionality with customized Apps, it is possible to push new communication patterns to a large number of devices. A main limitation of MANET is its connectivity. Messages can only be delivered within the connected topology. In sparse and mobile environments, this connectivity might get lost and causes a gap.

For less than a decade, *opportunistic networks* (OppNets) have been researched to fill this gap. An OppNet is a subclass of delay-tolerant networks (DTNs). In OppNet the connections are created over opportunity contacts so that an end-to-end path between the source and the destination node may never exist and even if it exists then for a short and unpredictable period of time. Long propagation and variable queuing delays may occur that lead MANET routing protocols to fail under such conditions. On the other hand, OppNet protocols enable devices to get interconnected with other regions devices by relaying messages in a store, carry & forward fashion. To have this store, carry & forward mechanism, the intermediate node must have a new protocol layer, the so-called bundle layer, in order to switch messages on top of heterogeneous (MANET) regions. All the nodes in the OppNet are entities, including the bundle layer, which can act as a host, a router or a gateway. By taking advantage of the nodes mobility data can be stored, carried and then be forwarded through the opportunistic contacts. Here, entire chunks of messages are transferred along a path from one storage node to another storage node with the expectation to reach their final destination. Thus, message routing is implemented in a dynamically unconnected/connected network.

Mobile ad hoc networks and opportunistic networks provide message-routing in connected- or dynamically partitioned networks which define only one part of a complete decentralization. In addition, fully-decentralized mobile applications require algorithm to self-manage the data-informations (e.g. which nodes are interesting as destination nodes) and the functionality of data-centric including to store and retrieve of information/data in the network. For fully decentralized and data-centric applications, *peer-to-peer networks* have been discussed in literature. Unstructured/structured overlays (like Gnutella, Chord or Kademlia) enable the participating nodes to publish, search or lookup documents of interest in a self-organized and fully decentralized fashion. Through the establishment of structured or unstructured P2P overlays on top of a MANET and an OppNet a common access to users and shared data is enabled. Thus, complex and fully decentralized application overlays can be built.

This work was partially supported by the BMWi Grant KF2846503KM4 - within the program Zentrales Innovationsprogramm Mittelstand (ZIM)

We believe that in future, based on mobile devices (e.g. smartphones) a combination of all three networks will emerge that comprise the following layers:

- Application layer: mobile devices running decentralized applications with a focus on node-to-node communication and distributed data storage.
- P2P layer: the application operates on top of a P2P network, in which nodes take equal roles to provide ID-based routing and distributed data storage.
- OppNet layer: the P2P overlay switches to an opportunistic network layer, based on dis-connectivity between various (wireless) communication islands.
- MANET layer: the P2P and opportunistic contacts are based on different wireless technologies, such as WiFi and Bluetooth, to create local MANETs.

The combination of these three networks defines various promising use cases which are elaborated next. Afterwards, various requirements for combined P2P, OppNet and MANET simulators are defined. In the fourth section the various simulators are introduced. In the last section, the perspective of an evaluation tool for these three networks is described. Finally, the paper ends with a conclusion.

II. Use Cases

MANETs and OppNets allow devices to communicate in geographically local areas through multi-hop routing within a connected wireless mesh network as well as through delay-tolerant protocols between islands of wireless sensor networks. P2P solutions will allow devices to organize relevant data and meta-data in this network and to solely use the available mobile nodes as storage points for storing and retrieving reliable data.

Sample use cases are characterized by two points: (1.) User-hosted files or services, without the explicit requirement of accessing the internet. This can be related to locally stored files or user-specific communication such as chatting or audio(/video) messaging that is possible with the smartphone app Firechat[5]). (2.) Geographically local communication, i.e. within the reach of MANETs or mobility range of users connected as MANET islands. Densely populated areas are very suitable locations for such applications.

One core area of applications of these combined networks are networks with strong bandwidth in local networks but only marginal bandwidth to remote areas or the internet. For example file sharing and synchronization between mobile devices can be beneficial by utilizing strong local ad hoc connections rather than using slow and expensive wireless cellular networks. Assuming a workshop in a remote location that includes a set of people who want to synchronize a shared folder on their smartphones with working documents. Through P2P file indexing and coordination on the data plane, suitable communications between their smartphones can be scheduled. Using MANETs, all suitable communication paths within the multi-hop range of the nodes can be identified and served. Once the nodes are synchronized, they can act as data ferries to either disseminate the data folders to a not directly reachable building for synchronization or for a delayed transfer of data to colleagues at their working environment.

Another use case can be the bandwidth-demanding application, such as interest-based wireless file-sharing, that is simple to realize by the mobility and the opportunistic exchanging and ferrying of data fragments. Assuming that users could share files of a given set of possible topics or even freely formulated new topics with their mobile devices. In addition, they can specify their interest in selected topics. The P2P mechanism could support this by indexing the provided files and facilitate the searching of files in a MANET. Then, the MANET layer would support the node-to-node communication for routing the announcement and searching queries of files as well the multi-hop file transfer. Finally, with the help of opportunistic routing protocols those files could be offered through further MANET islands. By the mobile nodes roaming between various MANET islands, it is possible to learn over time which kind of files/topics are offered and requested in the various MANET islands. If time and bandwidth permits then those nodes can act as data ferries. Over time, the users of such an application would collect interesting files over the day and enjoy the outcome of this mobile file-sharing application.

Disrupted connections or even attacks on the internet could be the for the unavailability of connectivity or services. For example, in recent revolutions and civil uproar - as the government faced with revolution and unwanted movements - the internet access just got turned off for a majority of the population [6]. In such a circumstance a robust decentralized communication application on top of MANET and OppNet is vital. These networks can enable local communication (e.g. encrypted chatting and messaging between users) or the interaction through data such as the distribution of stored mailing lists, blogs or organizational information. For example, minorities could organize through the sketched P2P, OppNet, and MANET networks for expressing their ideas to each other, for communicating in private, for drafting a common manifesto, or for organizing their views on the events which they are facing. The dynamics of these networks will allow (new) members (and their network nodes) to join as well as share and collect information without needing the internet. Enabling the people with technology to practice their rights to freedom of speech and to create and share data among interesting parties without the fear of oppression, is a great aim that can be reached with a fully decentralized mobile network.

These combined networks promote the decentralization of applications and thus aim to be the next evolutionary step of decentralized communication. One crucial step on this research path is to identify suitable protocols and their interdependencies. But before that, the requirements of a suitable and highly scalable evaluation tool has to be identified.

III. The Requirements for a Combined P2P, OppNet, and MANET Evaluation Tool

The focuses of this paper are simulation tools for (combined) protocols in the scope of P2P, OppNet, and MANET networks. While for each of these network types dedicated simulation environments exist, it remains what requirements are needed to find the best suitable simulators.

In the following, the functional and non-functional requirements for a simulation-environment for the evaluation of P2P/MANET/OppNet networks are defined:

- Availability of the simulator (Active): An active community should be available that uses and extends the

simulator continuously so its support is guaranteed for the next years.

- Event-based simulation core (Events): In contrast to round-based simulations, an event-based simulation allows to investigate the interdependencies between any actions in the simulation.
- Scalability protocols (Scale): The provision and support of large-scale networks that make it possible to simulate more than a thousand nodes so to achieve a scenario like a demonstration.
- Validated models (Models): Network characteristics models, such as application activity models, churn models, movement models, and node capacity models, should be validated and based on measurements.
- Support for multiple layers (Layers): An evaluation tool should consist of application, P2P, OppNet, and MANET layers and therefore it should support the layering of various protocols.

A set of basic protocols should be available for a simulation with multiple layers: (1.) Application: publish/subscribe mechanism (2.) P2P: structured and unstructured overlays such as Chord, Pastry, Kademlia, Gnutella etc. (3.) OppNet: Epidemic Routing, Spray and Wait, and Store, Carry and Forward. (4.) MANET: AODV, OLSR and DSR.

Based on those above requirements, various simulation tools are analyzed. The simulators are selected based on their usage in literature and the quotes in Google Scholar and no restrictions are made. In Table I an overview of the simulators are given. The symbol indication are as follow: "✓" for a good fulfillment of the requirement, "○" for a partial fulfillment of the requirement and a "×" for not fulfilling the specific requirement. In the following a detailed overview on P2P, OppNet and MANET simulators are given.

IV. Analysis of Evaluation Tools

Next, we present simulators for P2P, OppNet, and MANET and discuss their suitability to simulate all three networks.

A. Evaluation Tools for P2P Networks

Based on the overview of current and also outdated P2P simulators, the potential to be customized in the context of using OppNet and MANET is identified as well.

1) PeerfactSim.KOM: [7] is a discrete event-based, Java-written simulator that was developed at the Technical University in Darmstadt/Germany since 2006. Further developers are to be found at the University of Düsseldorf/Germany and Paderborn/Germany. The code is released under GPL license. It comprises well-structured components and layers which can be combined independently and it is extendable for simulating large variety of scenarios, without being dependent on a specific P2P architecture. PeerfactSim.KOM supports four different churn models and a great variety of structured and unstructured overlays like Chord, Pastry, Kademlia, CAN, Gnutella 0.4, Gnutella 0.6, and Gia. Furthermore, in PeerfactSim.KOM the possibility is given to assign to each host a geographical location (e.g. a specific city, country, or continent). Another feature of PeerfactSim.KOM is the packet scheduling model [31] as well as the mobility model that can simulate mobile devices [32]. It is quoted around 116 times, is actively maintained by a community, and can scale up to 10,000 nodes.

2) OverSim: [8] is a flexible P2P simulation framework, written in C/C++ and built on top of the discrete event-based simulation core OMNeT++ [19]. It is developed at the Institute of Telematics at the Karlsruhe Institute of Technology (KIT) since December 2006. OverSim comes with several structured and unstructured P2P protocols, namely Chord, Gia, Broose, Koorde, Pastry, Kademlia, Bamboo, Nice, Vast, NTree, Quon, and PubSubMMOG. To provide realistic simulations, which include dynamic user participation, OverSim offers four different churn models. It can be simulate over a simple underlay network model or real networks underlay but the possibility to simulate over a mobile network is not given. Oversim is quoted around 534 times. It is well documented, includins module- and C++-API documentation, and an installation and usage guide for the simulator. It scales up to 100,000 nodes.

3) PeerSim: [9] is a Java-based P2P simulator, which is mainly designed to support extreme scalability and dynamism. The simulator was started within the EU projects BISON and DELIS and has been released under the GPL open source license. Although PeerSim has been widely used, its latest version dates back to 2014. In order to provide high performance with more than 1,000,000 simulated nodes, the simulator uses simplified assumptions about the communication protocol stack for example by ignoring details of the transport layer. PeerSim can be used to simulate unstructured and structured overlays using cycle-based and event-based simulation models. It is quoted around 662 times. The documentation and the tutorials mainly focus on the cycle-based model, although the event-based model is more precise.

4) Overlay Weaver: [10] is an overlay construction toolkit in Java, developed in Grid Technology Research Center in Japan and initially released in January 2006 that aims to test new developed overlay algorithms and applications. Overlay Weaver only supports structured overlays, currently only six are implemented, scales up to approximately 350,000 simulated nodes and provides a common API for higher-level services (e.g. DHTs). Communication between nodes in Overlay Weaver can be emulated by passing around messages between different threads in one Java VM, by using real UDP or TCP, or by using real networks and multiple Java VMs on different hosts. Customizing the lower layers of the simulator (e.g. for MANET or OppNet support) is therefore difficult and inconvenient. Overlay Weaver is quoted 162 times. Although the website is kept up to date, the simulator description is incomplete and no API documentation is available.

5) ProtoPeer: [11] is a message-level, event-based P2P simulator written in Java which allows to reuse simulated code for real network applications without modifications. It was developed at the Swiss Federal Institute of Technology in Lausanne/Switzerland. For real network deployment, simulated timers and transport layer protocols can be exchanged by real timers or networking protocols. This allows the user to provide customized timer and network implementations so that even network models for MANETs might be used as layer below the overlay layer. The authors of ProtoPeer define the simulator's scalability limit at approximately 50,000 nodes by investigating simulations of lookups in Chord. ProtoPeer is quoted around 57 times. The project website of ProtoPeer, given in [11] was not accessible during our research.

Simulator	Citation	Active	Events	Scale	Models	Layers	Appl.	P2P	OppNet	MANET	Quotes	Link
Peer-to-Peer Simulators												
PeerfactSim.KOM	[7]	✓	✓	✓	✓	✓	✓	✓	×	✓	116	www.peerfact.org/
OverSim	[8]	✓	✓	✓	✓	○	✓	✓	×	✓	534	www.oversim.org/
PeerSim	[9]	✓	✓	✓	○	○	○	✓	×	×	662	peersim.sourceforge.net/
Overlay Weaver	[10]	✓	✓	×	×	×	○	✓	×	×	162	overlayweaver.sourceforge.net/
ProtoPeer	[11]	×	✓	✓	○	○	○	○	×	✓	57	protopeer.epfl.ch/
P2PSim	[12]	×	✓	×	×	×	○	○	×	×	67	pdos.csail.mit.edu/P2Psim/
PlanetSim	[13]	×	✓	×	×	×	×	○	×	×	191	projects-deim.urv.cat/trac/planetsim/
GnutellaSim	[14]	×	✓	×	×	×	×	×	×	×	127	www.cc.gatech.edu/computing/compass/gnutella/
GPS	[15]	×	✓	×	×	×	×	×	×	×	137	www.cs.binghamton.edu/∼nael/gps/
Opportunistic Network Simulators												
THE ONE	[16]	✓	✓	✓	✓	○	×	×	✓	×	1254	www.netlab.tkk.fi/tutkimus/dtn/theone/
DTN-Agent	[17]	○	✓	✓	✓	○	○	×	✓	×	14	www.spice-center.org/dtn-agent/
DTNSim2	[18]	×	×	○	✓	×	×	×	○	×	5	watwire.uwaterloo.ca/DTN/sim
Mobile Ad Hoc Network Simulators												
OMNeT++ with InetMANET	[19]	✓	✓	○	✓	✓	×	×	○	✓	2522	www.omnetpp.org/
NS3	[20]	✓	✓	✓	✓	○	×	×	×	✓	422	www.nsnam.org/
NS2	[21]	✓	✓	×	✓	○	×	×	×	✓	596	www.isi.edu/nsnam/ns/
GloMoSim	[22]	×	✓	×	○	○	×	×	×	✓	2324	pcl.cs.ucla.edu/projects/glomosim/
JiST/SWANS	[23]	×	✓	×	○	○	×	×	×	✓	63	jist.ece.cornell.edu/
GTNetS	[24]	×	✓	×	○	×	×	×	×	✓	467	www.ece.gatech.edu/research/labs/MANIACS/GTNetS/
J-Sim	[25]	×	✓	×	○	×	×	×	×	✓	410	sites.google.com/site/jsimofficial/
DIANEmu	[26]	×	✓	×	○	×	×	×	×	✓	18	www.ipd.uni-karlsruhe.de/DIANE/en/
QualNet	[27]	×	✓	×	○	×	×	×	×	✓	93	www.scalable-networks.com/content/
Jane	[28]	×	✓	×	○	×	×	×	×	✓	6	userpages.uni-koblenz.de/ unikorn/
NAB	[29]	×	✓	×	○	×	×	×	×	✓	-	http://nab.epfl.ch/
OpNet	[30]	×	✓	×	○	×	×	×	×	✓	370	www.opnet.com/

TABLE I. SIMULATOR OVERVIEW

6) P2PSim: [12] is a discrete event-based P2P simulator which is written in C++ and capable of simulating approximately 3000 nodes. It was developed at the Massachusetts Institute of Technology in 2004. In its latest state, P2PSim supports churn and node failures as well as the following overlays: Chord, Accordion, Koord, Kelips, Tapestry, and Kademlia. P2PSim is quoted around 67 times. Unfortunately, there exists no proper documentation for the usage of P2PSim.

7) PlanetSim: [13] is a simulation framework, written in Java, which aims to be easy for developing overlay services and algorithms. It was developed at the University "Rovira i Virgili" in Tarragona/Spain since 2004. In addition, it allows to run implemented code in network testbeds such as PlanetLab. Although PlanetSim has been designed to support structured and unstructured overlays, only Chord and Symphony are implemented. It is quoted 191 times. The website of PlanetSim is not accessible and therefore the PlanetSim is marked as inactive.

8) GnutellaSim: [14] is written in C++ and is a part of a simulation framework that is designed to be portable to different network simulators like ns-2 [21]. It was developed at Georgia Institute of Technology in 2003. GnutellaSim provides a common socket-like API between the P2P simulator and the used network simulator. The latest status of GnutellaSim dates back to September 2003 and provides only the implementation of Gnutella protocol version 0.4 and 0.6. It is quoted 127 times.

9) GPS: [15] is a message-level and event-driven P2P simulator which aims at the accurate modeling of P2P protocols. It was developed at the Binghamton University in 2005. It considers the underlying network by using macroscopic models to simulate the underlying TCP characteristics. GPS has mainly been used to simulate the BitTorrent protocol. However, it can be used to implement other structured or unstructured overlays and simple churn models. The simulator is abandoned, since its documentation and its website have not been updated for years now. It is quoted 137 times.

Conclusion: Considering the few actively maintained P2P simulators, only OverSim and PeerfactSim.KOM match a majority of our basic requirements and are considered as potential candidates. Oversim has all the needed P2P overlays, but it does not have MANET and OppNet models. On the other hand, PeerfactSim.KOM seems to be more reasonable due to the support of multi-layer investigation including a mobility network model.

B. Evaluation Tools for Opportunistic Networks

The research field of opportunistic networks is quite new and only a few simulators have been introduced.

1) Opportunistic Network Environment (ONE): [16] is a Java-based discrete event simulator targeted to research Delay Tolerant Networks and its subclasses. This simulator has been developed in the SINDTN and CATDTN projects supported by Nokia Research Center (Finland) starting in 2008. It is well maintained and regularly used in publications. The ONE simulator allows researchers to create scenarios for the OppNet based on different synthetic movement models. While it comes with six sophisticated built-in routing protocols, its application models are quite simple. Besides simple high-level application models (unicast and request-response interactions) it also supports an application-layer framework. For the simple application models only message source and destination information as well as message sizes and intervals are considered. The content of the message is not considered relevant. A second limitation is the link layer abstracting to bit-rate and range. However, the simulator supports generic interfaces that can be used to model nodes with multiple radios (e.g., Bluetooth, WiFi). By using the application framework it is possible, with some circumstance, to build a P2P layer or a more complex (data-centric) application on top. Supporting

MANET, in the lowest layer, requires drastic changes in order to fit to the suggested requirements. It is quoted 1254 times.

2) DTN Agent: [17] is implemented as add-on module for the ns-2 simulator [21] to deploy DTN behavior over the traditional TCP/IP stack. The latest version released was in 2012. This agent was supported by Space Internetworking Center (SPICE) founded on 2010. It gives the ability for researchers to explore and focus on issues like routing algorithms for opportunistic networks, bundle fragmentation, and storage management. The focus of this module is on static routing mainly for space internetworking, it does not support dynamic topology changes like MANETs. The simulator does not scale well due to its considered scenario. It is quoted 14 times.

3) DTNSim2: [18] is a discrete event simulator written in Java especially for research on Delay Tolerant Networks (DTNs), the main focus of this simulator is on DTN routing protocols, based on the store, carry & forward fashion. The routing protocols implemented are epidemic. The simulator scales only to a few hundred nodes (e.g. in [33] only 30 nodes have been simulated). The only available version was released in 2006, which leads to the conclusion that the project has been abandoned. It is quoted 5 times.

Conclusion: With respect to the few simulators, only ONE is the actively maintained simulator in this field. However, due to its limitations to support higher level applications and an accurate mobile networking layer, it requires large effort to extend ONE to support P2P, OppNet, and MANET all together.

C. Evaluation Tools for MANETs

Next, the simulators for MANETs are described and their ability to support OppNets and P2P networks are investigated.

1) OMNeT++ with INETMANET Network Model: [19] is a discrete-event framework written in C++ and atop of OMNeT++. It was first introduced by György Pongor in 1993. OMNeT++ provides a simulation-core and utility-classes for developing overlay and underlay simulations such as P2P or MANET. The topologies can be created either by using simple text (with OMNeT++'s own language NAD) or by using the provided GUI. The IDE-Environment is built atop of Eclipse. For OMNeT++ a special framework, called INETMANET, is provided that supports several MANET routing protocols such as AODV, GPSR, DYMO and many more. INETMANET is created by Alfonso Ariza at the University of Malage and it is continuously updated. However, our investigations showed that the overlay node model of Oversim and the INETMANET network model are not combinable. To run a P2P network on top of a MANET with the combination of these simulators (OverSim & INETMANET in OMNeT++) would mainly require a reimplementation of one of the models from one of those simulators. The simulator is actively maintained and regular conferences are held for the community.OMNeT++ is quoted around 2522 times.

2) ns-3: Network Simulator version 3: [20] is a discrete-event simulator written from the scratch completely in C++. It was developed by a team led by Tom Henderson at the University of Washington in 2006 with the objective to replace ns-2 [21]. Its main focus is on wireless networks and it supports common routing protocols such as AODV and OLSR. Another feature of ns-3 is the real time scheduler for sending and receiving packages with real devices. By using a distributed scheduler the scalability of ns-3 was tested in [34] and the simulator was capable to simulate 360,448,000 nodes with 176 computers. Ns-3 has a large community of users and developers. The release-period for a new stable version is every three months. Its main limitation lies in the lack of different networking layer implementations such as OppNets or P2P networks. It is quoted approximately 422 times.

3) ns-2: Network Simulator version 2: [21] is a discrete-event simulator. The core of the simulator is written in C++ and for the simulation scenarios the script-language OTcl is used. Ns-2 was developed by Steve McCanne in 1996. It's origin supports only wired networks but with some extension the possibility is provided to simulate wireless networks. The simulator suffers from low scalability i.e. not more than few hundred nodes can be simulated. In 2006 it was one of the most popular simulators [29]. The latest version dates back to 2011 and it is quoted approximately 596 times.

4) Global Mobile System Simulator (GloMoSim): [22] is a discrete-event simulator written in a C-based language called Parallel Simulation Environment for Complex Systems (PARSEC). It was developed at UCLA in California/USA in 1998/1999. GloMoSim uses for each node a protocol-stack that consists of multiple layers and each layer represents a different protocol of the wireless communication. The communication between each of this layers can be established via a special API. It was the second most popular simulator back in 2006 [29], but unfortunately the project is not anymore active and no website is available. It is quoted around 2324 times.

5) Scalable Wireless Ad hoc Network Simulator (SWANS): [23] is a Java-based discrete-event simulator built atop Java in Simulation Time (JiST). It was developed at the Cornell University in 2004. A special feature of SWANS is that standard Java applications, such as web servers and peer-to-peer applications, can be run over the simulator without any adaptation. Its main focus is on MANET and supports protocols such as DSR, AODV and AntSec [35]. SWANS is a large-scale simulator that supports up to 1,000,000 nodes. It is abandoned in the year 2005 and it is quoted around 63 times.

6) Georgia Tech Network Simulator (GTNetS): [24] is a parallel discrete event simulator written in C++. It was developed at GeorgiaTech Institute (Atlanta, USA). GTNetS aims at large-scale simulations and protocol stacks that are very close to real world protocol stacks. For the connection between the protocol stacks, system calls similar to POSIX, such as connect, listen, send and sendto, are used. It includes the IEEE 802.11 standard which allows to simulate wireless networks. MANET routing protocols, such as AODV and DSR, as well as P2P overlays, such as Gnutella and GCache, are implemented on GTNetS. Although GTNetS is being frequently used in literature it has been abandoned with its latest version in 2008. It is quoted around 467 times.

7) J-Sim - JavaSim: [25] is written in Java and was developed at Ohio State University together with Illinois Universities. This simulator is built on top of the autonomous component architecture. The feature here is that each entity, such as nodes, routers, or networks, becomes a component that has its own input- and output-sockets. This entity can be a composition of components, such as a network component, which contains routers components and node components. While initially it was designed for wired networks, the support for IEEE 802.11 was added in order to simulate the physical

behavior of mobile devices. Since 2008 the simulator is abandoned and it is quoted around 410 times.

8) DIANEmu: [26] is an event based simulator written in Java. It was developed at the University of Karlsruhe/Germany in 2003. The aim of the developers was to simulate MANET protocols in the application layer of the OSI Model instead of the lower layers. The simulator consists of five main layers, namely the connectivity, network, protocol, user and meta layer. The simulator is not publicly available and suffers from weak documentation. It is quoted approximately 18 times.

9) QualNet: [27] is the commercial version of GloMoSim. It is maintained by the company SCALABLE Network Technologies in Cilver City/USA. It has a user friendly GUI, useful documentation, and a lot of routing protocols such as AODV, LAR, ZRP, and so on. It is still maintained by SCALABLE Network Technologies and it is quoted 93 times.

10) Java Ad hoc Network Environment (Jane): [28] is in Java written simulator developed in 2006 at the University of Trier/Germany. In Jane all the entities, such as routing protocols and network layers, are services which communicate via event-based messaging services. Jane also acts like an operating system in which service management, event execution management and time out management are handled. The webpage is not available anymore and there are no documentations for the interfaces, implementations, and scalability of the simulator on the internet. It is quoted 6 times.

11) Network In A Box (NAB): is a discrete-event simulator written in OCaml, a language of the ML programming language, developed at EPFL Lausanne in Switzerland. The goal of this simulator was to be scalable, to support visualization, and to have a realistic mobile environment. The webpage is not available anymore and there are no documents or surveys that can provide further information. There is no paper published, regarding this simulator, from the EPFL and for that reason the citation quote is dismissed here.

12) Optimized Network Engineering Tools (OpNet): [30] is a commercial simulator written in C++. It was developed at the Massachusetts Institute of Technology in 1986 and is now a part of the company Riverbed. In 2006, it was one of the most popular commercial simulators [29]. It has special library models that allows to modify existing network models or to develop new ones. Currently the webpage, including the documentation or downloads options, is not available on the internet. It is quoted approximately 370 times.

Conclusion: In the survey of MANET simulators many abandoned simulators are found, with only OMNeT++ with INETMANET and ns-3 actively maintained. While the popularity of ns-2 is still high, on the long term it will be replaced by ns-3. All simulators in this field are lacking of protocols for P2P and OppNets and many as well options for implementing further layers atop of the MANET layer. Since INETMANET is built atop of the OMNeT++ simulator, it can be assumed that INETMANET can be combined with Oversim in order to create a combined P2P, OppNet and MANET simulator. Based on exhaustive investigation the conclusion is that this combination is very difficult to implement due to the incompatible underlying network module characteristics and severe implementation efforts are required for that.

V. Perspective for an Evaluation Tool for P2P, OppNet and MANET Networks

In order to identify the most suitable basis simulator to feature the simulation and evaluation of combined P2P, OppNet, and MANET networks, this paper takes a look at the most suitable simulators of those three network categories.

Considering the introduced P2P simulators, only PeerfactSim.KOM and OverSim are still actively maintained and have a large community. OverSim has been built on top of the OMNeT++ simulation framework and for that reason it can be assumed that OverSim and INETMANET framework can be combined together with less effort. But after an exhaustive research, the result is that both frameworks are using different communication modules. Additionally, the way that these modules are connected to other models, within the simulator, are not identical. In Oversim, the TCP and UDP modules are connected directly to a module called Overlay. In INETMANET these protocols are connected directly to the network module. Based on these differences and the complexity of those modules the combination of these simulators is only feasible with a large effort.

PeerfactSim.KOM, on the other hand, profits from its scalability, the variety of supported layers and applications that benefits from the expandability of its different churn and network models. In addition, PeerfactSim.KOM supports the ability to simulate the separation of any geographical region from the entire network and simple mobility model as shown in [32]. What is missing here are suitable MANET and OppNet protocols within these supported multi-layer simulation framework. Implementing prominent protocols for MANET and OppNet seems rather doable (as early MANET and OppNet protocols are not too complex) and therefore PeerfactSim.KOM is our preferred candidate.

Regarding the simulations of OppNets only the Opportunistic Network Environment (ONE) is actively maintained and a promising candidate for deeper investigations. The other two OppNet simulators focus on delay-tolerant networks (e.g. for satellite communications) and thus they do not scale to desired network sizes. Nevertheless, ONE only provides simple application models in which message contents are not considered to be relevant. Moreover, lower layers are massively abstracted so that changes in the link layer or network layer cannot be done easily. For those reasons, it would require too much effort to extend the ONE-Simulator for supporting the P2P, OppNet, and MANET combined together.

Out of the introduced simulators for MANETs only ns-3 and OMNeT++ with INETMANET seems to be the possible candidates. Ns-3 supports a great variety of scenarios, since they provide low level simulation and possibilities to create any applications on the top of the available layers. This simulator supports only MANET and they are currently no modules for simulating OppNets and P2P systems. The implementation of these networks seems to be feasible and therefore ns-3 is our second candidate for a combined P2P, OppNet, and MANET simulator. OMNeT++ with INETMANET supports MANET but does not provide a P2P and OppNet implementation. And as previously mentioned, a combination of Oversim with OMNeT++'s INETMANET is only feasible with a large effort and therefore OMNet++ is excluded as a potential candidate.

VI. Conclusion

The operation of P2P networks with the capability to switch to an opportunistic networks on top of a MANET is an emerging research field motivated through the vast amount of decentralized, powerful, and MANET-capable devices, such as smartphones, including the demand for local and undisrupted communication. For this field of research a survey on simulation tools is established with the objective to find a suitable evaluation environment that is capable to support all three types of networks at once. While for P2P, OppNet, and MANET several types of simulators exist, none of these simulators support a combination of all three network types. Considering these different simulators with these three different networks only a few simulators seems to fulfill some of the demanded requirements that is described in Section III. However, none of them supports these networks at once.

From our point of view, PeerfactSim.KOM [7] seems to be the most suitable simulator for the integration of these combined networks. Its main benefit is the support of various network layers. P2P network elements such as overlays, data management mechanisms, monitoring overlays, and sophisticated applications are already supported. Furthermore, with [32], the mobility part and the basis for MANETs are set. What remains to be done, in order to create a multi-network simulator, is the implementation of an OppNet layer with suitable protocols. On the other hand, the ns-3 qualifies as a potential candidate as well. But here both the OppNet and the P2P protocols have to be implemented first. All in all, the next step will be therefore either to extend the PeerfactSim.KOM with a suitable OppNet protocol or to extend ns-3 with OppNet and P2P protocols. As a consequence, a simulation tool can be created in which the possibility to implement and evaluate P2P, OppNet, and MANET protocols is given. Such combined networks can provide local, secure, and decentralized communication and data handling in cases of sever attacks on the internet infrastructure, such as during the Arab Spring.

References

[1] C. Perkins, E. M. Belding-Royer, S. Das *et al.*, "RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing," *Internet RFCs*, 2003.

[2] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Comp.* Springer, 1996, pp. 153–181.

[3] T. H. Clausen and P. Jacquet, "RFC 3626: Optimized Link State Routing Protocol (OLSR)," *Internet RFCs*, 2003.

[4] P. S. Mogre, K. Graffi, M. Hollick, and R. Steinmetz, "AntSec, WatchAnt, and AntRep: Innovative Security Mechanisms for Wireless Mesh Networks," in *Proc. of Int. Conf. on Local Comp. Networks*, 2007.

[5] O. G. I. Benoliel, "Firechat," http://www.firech.at, 2014.

[6] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé, "Analysis of Country-wide Internet Outages caused by Censorship," in *Proc. of the ACM SIGCOMM Conf. on Internet Measurement*. ACM, 2011, pp. 1–18.

[7] K. Graffi, "PeerfactSim.KOM: A P2P System Simulator - Experiences and Lessons Learned," in *Proc. of the Int. Conf. on Peer-to-Peer Computing*, 2011, pp. 154–155.

[8] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Scalable and Flexible Overlay Framework for Simulation and Real Network Applications," in *Proc. of the IEEE Int. Conf. on Peer-to-Peer Comp.*, 2009, pp. 87–88.

[9] A. Montresor and M. Jelasity, "PeerSim: A Scalable P2P Simulator," in *Proc. of the Int. Conf. on Peer-to-Peer Com.* IEEE, 2009, pp. 99–100.

[10] K. Shudo, Y. Tanaka, and S. Sekiguchi, "Overlay Weaver: An Overlay Construction Toolkit," *Computer Communications*, vol. 31, no. 2, pp. 402–412, 2008.

[11] W. Galuba, K. Aberer, Z. Despotovic, and W. Kellerer, "ProtoPeer: Distributed Systems Prototyping Toolkit," in *Proc. of the IEEE Conf. on P2P Computing*, 2009.

[12] T. M. Gil, F. Kaashoek, J. Li, R. Morris, and J. Stribling, "P2Psim: A Simulator for Peer-to-Peer Protocols," http://pdos.csail.mit.edu/p2psim/, May 2005.

[13] P. G. López, C. Pairot, R. Mondéjar, J. P. Ahulló, H. Tejedor, and R. Rallo, "PlanetSim: A New Overlay Network Simulation Framework," in *Proc. of the Int. Workshop on Software Engineering and Middleware*. Springer, 2004, pp. 123–136.

[14] Q. He, M. H. Ammar, G. F. Riley, H. Raj, and R. Fujimoto, "Mapping Peer Behavior to Packet-level Details: A Framework for Packet-level Simulation of Peer-to-Peer Systems," in *Proc. of the IEEE Int. Symp. on MASCOTS*. IEEE, 2003, pp. 71–78.

[15] W. Yang and N. B. Abu-Ghazaleh, "GPS: A General Peer-to-Peer Simulator and its Use for Modeling BitTorrent," in *Proc. of the Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecomm. Sys.* IEEE, 2005, pp. 425–434.

[16] A. Keränen, T. Kärkkäinen, and J. Ott, "Simulating Mobility and DTNs with the ONE," *Journal of Comm.*, vol. 5, no. 2, pp. 92–105, 2010.

[17] D. Vardalis and V. Tsaoussidis, "DTN Agent for ns-2," http://www.spice-center.org/dtn-agent/, 2010.

[18] WatWire Research Group, "DTNSim2," http://watwire.uwaterloo.ca/DTN/sim/, 2006.

[19] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *Proc. of the Int. Conf. on Simulation Tools and Techniques for Communications*. ICST, 2008.

[20] ns3 Project, "The Network Simulator - ns-3," http://nsnam.org, 2006.

[21] UC Berkeley and LBL and USC/ISI, and Xerox PARC, "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns/, 1993.

[22] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks," in *Workshop on Parallel and Distr. Sim.*, 1998.

[23] R. Barr, "Swans-Scalable Wireless Ad hoc Network Simulator," *User Guide*, 2004.

[24] G. F. Riley, "The Georgia Tech Network Simulator," in *Proc. of the ACM SIGCOMM Workshop on Models, Methods and Tools for Repr. Net. Research*, 2003.

[25] A. Sobeih, J. C. Hou, L.-C. Kung, N. Li, H. Zhang, W.-P. Chen, H.-Y. Tyan, and H. Lim, "J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks," *Wireless Comm.*, vol. 13, no. 4, 2006.

[26] M. Klein, *DIANEmu: A Java based Generic Simulation Environment for Distributed Protocols*. Univ. Karlsruhe, Fak. für Informatik, 2003.

[27] SCALABLE Network Technologies, "QualNet," http://web.scalable-networks.com/content/qualnet, 2014.

[28] D. Görgen, H. Frey, and C. Hiedels, "JANE–A Simulation Platform for Ad Hoc Network Applications," in *Proc. of the IEEE Int. Symp. on MASCOTS*, 2006.

[29] L. Hogie, P. Bouvry, and F. Guinand, "An Overview of MANETs Simulation," *Proc. on the Electronic Notes for Theoretical Computer Science*, vol. 150, pp. 81–101, 2006.

[30] Riverbed, "Opt. Net. Eng. Tools (OpNet)," http://opnet.com/, 1986.

[31] K. Graffi, K. Pussep, S. Kaune, A. Kovacevic, N. Liebau, and R. Steinmetz, "Overlay Bandwidth Management: Scheduling and Active Queue Management of Overlay Flows," in *IEEE LCN '07: Proc.of the Int. Conf. on Local Computer Networks*, 2007.

[32] D. Stingl, B. Richerzhagen, F. Zollner, C. Gross, and R. Steinmetz, "PeerfactSim.KOM: Take it back to the Streets," in *Proc. of the Int. Conf. on High Performance Computing and Simulation (HPCS)*, 2013.

[33] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical Routing in Delay-Tolerant Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, 2007.

[34] K. Renard, C. Peri, and J. Clarke, "A performance and scalability evaluation of the ns-3 distributed scheduler," in *Proc. on the SimuTools*. ACM, 2012, pp. 378–382.

[35] P. S. Mogre, K. Graffi, M. Hollick, and R. Steinmetz, "A Security Framework for Wireless Mesh Networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 3, pp. 371–391, 2011.