SAROS: A social-aware opportunistic forwarding simulator

Soumaia Al Ayyat
soumaia@aucegypt.edu

SAROS: A Social-Aware Opportunistic Forwarding Simulator

Soumaia Al Ayyat
CS and Eng. Dept.
The American University in Cairo
Email: soumaia@aucegypt.edu

Sherif G. Aly
CS and Eng. Dept.
The American University in Cairo
Email: sgamal@aucegypt.edu

Khaled A. Harras
School of Computer Science
Carnegie Mellon University
Email: kharras@cs.cmu.edu

***Abstract*—Many applications are being developed to leverage the popularity of mobile opportunistic networks. However, building adaptive testbeds can be costly and challenging. This challenge motivates the need for effective opportunistic network simulators to provide a variety of opportunistic environment setups, and evaluate proposed applications and protocols with a comprehensive set of metrics. This paper presents SAROS, a simulator of opportunistic networking environments with a variety of interest distributions, power consumption distributions, imported real traces, and social network integration. The simulator provides a wide variety of evaluation metrics that are not offered by comparable simulators. Finally, SAROS also implements several opportunistic forwarding algorithms ranging from social-oblivious algorithms to interest and power-aware social-based algorithms.**

I. Introduction

With the growing advancement in network technology and the popularity of social media, 40% of the world population has an Internet connection with traffic reaching 686 billion GB [1]. Moreover, the pervasiveness of mobile devices and their progressive smart features motivate users to view and share content on social media, contributing to the production of massive traffic data volume. There are 1.55 billion monthly active Facebook users among which 1.39 billion are active mobile users [2], and people daily watch hundreds of millions of video hours on YouTube, generating billions of views [3].

Despite all these advantages, there is a set of challenges that impede users' appreciation of the offered services. Among these obstacles is mobile data demand that exceeds infrastructure support. This bandwidth demand-supply gap causes frequent contention due to the overwhelming uploads of media content and network access from pervasive mobile devices. With such increasing mobile data traffic, the network infrastructure becomes overloaded and users experience occasional network service unavailability, in addition to the rising service delivery cost [4] which may discourage them from engaging in many of the offered services. Moreover, not all people have predefined routes connecting them, and not all places are covered by the available network infrastructure [5]. All these challenges raise a call for ad hoc connections [6], delay tolerant connections [7] [8] [9] and opportunistic networks [10] [11] for communications in such challenged environments.

Reliance on content delivery via mobile opportunistic networks calls for state-of-the-art opportunistic forwarding algorithms [12] [13] [14] [15] [5] to complement delivery via established network infrastructure. However, practically assessing and evaluating these algorithms in such domains is extremely challenging since establishing realistic testbeds is both time consuming and costly. Reproducing real traces is also difficult due to external environment challenges [16] [17]. On the other hand, evaluation through synthetic mobility models allows for fine-tuning but covers limited mobility characteristics [17]. We believe that building realistic simulators that reflect current usage trends and state of the art solutions is the compromise; simulators facilitate the examination of various setups in controlled and reproducible environments.

There are several well-known simulators in the area of opportunistic networks such as ONE [17] and OMNeT++ [16]. These simulators provide measures of the standard network-based performance metrics, as well as a graphical interface for parameter input and animated output [18]. However, these simulators fall short in incorporating a set of features such as generating or importing social graphs; synthesizing or importing users' interest vectors; simulating power consumption; simulating realistic usage profiles; fully implementing power consumption patterns of mobile nodes; and implementing several state-of-the-art social-aware opportunistic forwarding algorithms, against which researchers can compare their solutions.

This paper proposes *SAROS*, a **S**ocial **A**wa**R**e **O**pportunistic Forwarding **S**imulator, developed to simulate mobile opportunistic networking environments. Our goal is to implement an opportunistic network simulator to facilitate evaluating the performance of novel social-aware forwarding algorithms. This simulator is enriched with realistic mobility model implementations [19] and manipulation of real mobility traces such as SIGCOMM09 [20] and INFOCOM06 [21] mobility traces. For realistic interest and social-aware simulations, SAROS encompasses synthesized social and interest graphs as well as real data from realistic settings. For realistic power simulations, SAROS also imports real power consumption models of popular mobile brands [22], imports real usage profiles [23], and incorporates the modified kinetic battery model [24].

SAROS implements state-of-the-art social-based opportunistic forwarding algorithms such as ProfileCast [13], BubbleRap [25], SocialCast [12] and PeopleRank [14]. It also modifies them to be social and power aware. Additionally, SAROS provides a sophisticated interface and produces a rich set of output data and graphs of measured metrics. Finally, we have conducted a set of experiments to validate and verify the implementation of this simulator.

978-1-4673-9814-5/16/$31.00 ©2016 IEEE

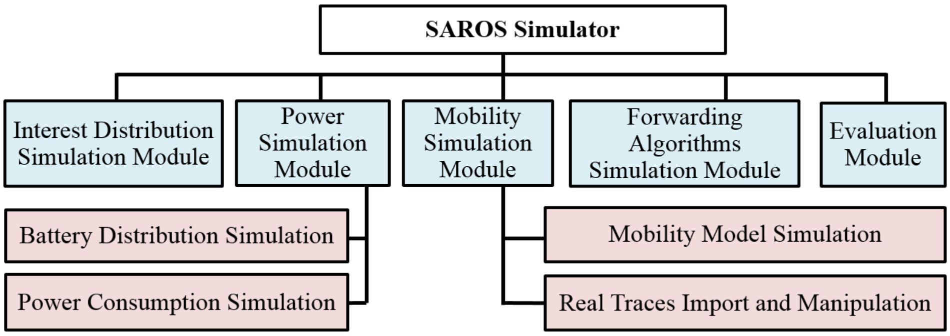

Fig. 1. SAROS Simulator Modular Architecture

II. Related Work

Opportunistic network simulators are categorized according to several criteria [18]; namely, stochastic or deterministic; steady state or dynamic; terminating or non terminating; discrete, continuous or hybrid; and local or distributed. Our area of interest is in discrete-event simulators. There are several well-known discrete-event opportunistic networks simulators such as ONE [17], NS2 [18], OMNeT++ [16] and OPNET [26] that offer significant features such as: **1)** a graphical user interface for the initial configuration of topology setup and parameter settings; **2)** support for modeling different network-specific hardware; and **3)** an interface to model, graph, and animate resulting output [18] [27].

However, these simulators fall short in simulating the following: **1)** Social-aware features such as generating or importing social graphs; **2)** Interest-aware features such as synthesizing or importing interest vectors, or considering the significance of the intermediate nodes' interest as an incentive to forward messages; **3)** Power-aware features such as simulating power consumption, simulating real usage profiles, and fully implementing the power consumption patterns of mobile nodes[1] [18]; **4)** Implementation of many state-of-the-art social-aware opportunistic forwarding algorithms such as SocialCast [12] and ProfileCast [13]; **5)** OPNET and NS2 also suffer from the object-oriented scalability problems [18]; **6)** The Packet formats, energy models, MAC protocols, and sensing hardware modelled in NS2 all differ from those found in most wireless devices [18]; and **7)** Import of real mobility traces and mobility models except for the opportunistic networking extension of OMNeT++ [16]. However, it is worth mentioning that ONE simulator allows the generation of node movement from an external program or from a real-world GPS trace, but users have to convert trace files into a format suitable for the External Movement module.

In this research paper, we present SAROS, an event-driven opportunistic network simulator, that contributes in overcoming many of these drawbacks.

III. Modular Architecture

The SAROS simulator, as its architecture shows in Figure 1, is constructed from a set of modules simulating content forwarding in a mobile opportunistic network within a certain area: **1)** An *interest distribution simulation module* simulating interest information stored on users' mobile devices; **2)** A *power simulation module* simulating the mobile devices' power distribution and consumption; **3)** A *mobility simulation module* that simulates user mobility, their encounters with other users along their path, and also enables importing a set of real mobility traces accompanied with their interest and social graphs; **4)** A *forwarding algorithms simulation module* implementing a variety of opportunistic forwarding algorithms to simulate message delivery; and **5)** An *evaluation module* that implements a set of performance metrics. These modules are detailed in the remainder of this section.

[1]The ONE simulator has an add-on for implementing general energy support while the battery module of the ZigBee OPNET Simulation Model computes the consumed and remaining energy levels as per the MICAz/TelosB mote specifications only [28].

A. Interest Distribution Simulation Module

To simulate the variance in users' interest in a certain message, SAROS sets thresholds for the similarity interest, which is the similarity between the interest vector of both the user and the message. The similarity interest of the sender acts as a knob controlling the acceptable set of contacted uninterested users since it acts as a starting cutoff point for forwarder selection. Thus, it is a parameter in the simulator.

The interest distribution simulation module imports interest distributions within the real mobility traces and synthesized interest distributions as follows: ***1) Normal Interest Distribution:*** The simulator distributes users according to a discrete normal interest distribution where the destination set covers 2% of the community, the interested forwarders set covers 48%, while the remaining 50% are uninterested nodes. ***2) Discrete Uniform Distribution:*** The simulator equally distributes users among 11 categories with varying interest levels in the range [0-1]. Accordingly, the destination set constitutes 18% of the mobile users' population while the interested forwarders cover 36%. The remaining 46% of the population are uninterested. ***3) Two Disjoint Interest subgraphs:*** The simulator divides users into two separate interest graphs: 20% of the community are destination nodes while the remaining 80% of the community are uninterested nodes. ***4) Location-based Interest:*** To simulate location-based interest distribution, the simulator is given the coordinates of a certain area within the simulation which is considered the starting place for all interested users.

B. Power Simulation Module

The power simulation module generates realistic power consumption values and various battery level distributions upon which the simulation runs are based. This module generates the initial distribution of battery levels before the simulations start. It also implements the battery discharge and charge recovery model, imports the usage profiles of mobile users, and randomly distributes these profiles among the mobile nodes. Finally, the module simulates various alternatives of battery depletion rates.

Initial Battery Distributions: Battery distributions are simulated for various purposes. ***1) Full Battery Distribution:*** All mobile nodes start with full battery levels to extract the effect of each algorithm on consuming the nodes' power. ***2) Normal Battery Distribution:*** The mobile nodes' battery levels follow a discrete normal battery distribution to resemble battery communities in real life. The result of this distribution is stored in an external file for later use in case of running in batch mode instead of the interactive menu. ***3) Heatmap Battery Distribution:*** The simulations may use a distribution that is based on a real dataset of the remaining battery capacity recorded by [29] for 10 mobile nodes in 24 hours. These recorded values are randomly distributed among the nodes simulated at the beginning of the simulation runs.

Battery Discharge Simulation: The simulator models the discharge behavior of the batteries of the simulated mobile devices. There are two options in this simulation: ***1) The Simple Battery Model*** that does not consider recovery effect of the Li-ion battery. It is rather a simple charge deduction from the charge capacity of the battery upon any wireless communication such as scanning nodes, forwarding or receiving messages. In the idle state, the battery does not recover any charges. ***2) The Kinetic Battery Model*** (KiBaM)[24] is a well-known analytical model of the Ni-MH battery charges' chemical kinetic process. It models the battery as two wells of charge; namely, the available-charge well which supplies electrons directly to the load, and the bound-charge well which supplies electrons only to the available-charge well. SAROS implements the stochastic modified KiBaM model [24] that models the battery charges in a 2-charge-well model and considers the recovery and rate capacity effect for batteries with the following parameters: the total charge of the battery, the capacity ratio between the two charge wells, the two wells' heights and the battery conductance.

Mobile Users' Usage Profiles: Several usage patterns are simulated and randomly assigned to users. The simulator applies usage profiles defined and recorded in another research work [23] where the authors study and categorize energy usage and battery lifetime under five usage profiles: Suspend, Casual, Regular, Business, and Portable Media Device.

Mobile Brands' Power Consumption: The simulator imports the power consumption values of four popular phone brands studied in [22]. These values are used within the power consumption calculations for the modes of Wifi/bluetooth scanning, forward/receive, and idle.

Depletion Rate Distributions: The simulator implements the following depletion rate distributions: ***1) Real Dataset Depletion Rate:*** In the real battery distribution dataset environment, the depletion rate is computed based on the difference between the given battery distribution at the beginning and at the end of the hour. ***2) Usage Profile-based Depletion Rate:*** The depletion rate is calculated based on the power consumption per usage profile imported from [23]. ***3) Random Depletion Rate:*** The simulator randomly and uniformly distributes the depletion rate among the nodes in the range of [0% - 100%].

C. Mobility Simulation Module

This module imports one mobility model, and also imports and manipulates various real mobility traces. These real traces are recorded from a mall environment, two conference environments, and a university campus environment.

Self-similar Least Action Walk (SLAW) [19] is a mobility model for human walks that produces synthetic walk traces. SLAW Trace Generator is a MATLAB-based program which produces mobility traces effective in representing social contexts present among people sharing common interests or those in a single community such as university campus, companies, and theme parks. We have made a small modification in the internal parameters of the generator to make it produce detailed traces per second instead of per-minute traces.

Import of Real Mobility Traces: The simulator imports several real datasets, but has to manipulate their data format to fit that accepted by the simulator. Here are the details and manipulation process of the imported datasets.

SIGCOMM09 Dataset: To further validate the algorithms' performance using real social-based mobility traces, SAROS imports the mobility traces, interests, and friendship graphs gathered during the SIGCOMM 2009 conference [20]. In this conference, 76 participants were handed smart phones and asked to use the installed MobiClique application for mobile social networking during the conference. Their social information; namely, the list of friends and interests, was collected from their Facebook social profile. Thus, this dataset provides real social information of the 76 participants, and the encounter traces of 4700 users with no social information. SAROS can extrapolate, with acceptable precision, the social and interest profiles to cover all the detected users based on the program schedule of the conference, the map of the session rooms, and the mobility traces. Finally, the simulator randomly picks some users to be the message senders.

INFOCOM06 Dataset:The simulator imports real mobility traces gathered during the INFOCOM 2006 conference [21]. In that experiment, 20 static iMote nodes were installed to detect mobile devices within a 100m range, and 78 conference attendees were given iMotes with a 30m range. 4704 nodes were detected with very few who filled the questionnaire and explicitly mentioned their interest. SAROS imports these traces and synthetically generates user profiles, friend lists, and interest feature vectors for the detected users. In manipulating the conference data, SAROS considers the 20 locations to be equivalent to 20 areas of interest. The simulator has computed the frequency of residence of the 4704 users in each of the 20 locations, and used it to compute user interest in each track. After extra refinement made by the simulator, we found that 75.78% of the users have not been detected at any of the 20 locations; thus, these users were not interested in any of the 20 interest areas. Further manipulation through extrapolating interest based on user mobility, the program schedule, and the sessions' locations enables deducing the remaining users' interest. To construct a friendship graph, SAROS considers any two users to be friends if they share interest in at least *m* of the 20 locations. Finally, the message sources' locations are picked from the static nodes' locations.

MallEnv Dataset: The simulator imports and manipulates real encounter traces in a shopping mall environment that were collected and published by other researchers [30]. The experiment was held by placing static mobile devices in 13 shops. Also, 7 shop employees in some of these shops were handed mobile devices. The experiment lasted for 6 days to collect the timestamp and location of encounters with any mobile device that came within the 30m range of Bluetooth. The simulator imports the traces and converts them into 1-day trace files. These trace files are then used within the simulator to extract the required number of hours of mobility traces for use within the simulation runs.

St. Andrews University Dataset: This is a traceset of a privacy study, including encounters, sharing preferences, and accelerometer readings conducted in two universities [31]; two separate runs were performed at the University of St. Andrews, while another two runs were performed at University College London with 20 students participating in each run. The mobile phones handed to the participants automatically collected the users' locations and uploaded them to a server. Participants could choose the information to be disclosed on Facebook, and to whom it could be disclosed. Thus, The simulator imports this dataset as it represents a university environment with students' mobility traces and social networks.

D. Forwarding Algorithms Simulation Module

The forwarding algorithms module implements a wide variety of opportunistic forwarding algorithms. These algorithms vary from social-oblivious forwarding to interest-and-power-aware opportunistic forwarding algorithms.

1) The **Wait Destination** algorithm is a social-and-power-oblivious opportunistic algorithm that relies on the source node in delivering the message to the destination node. That is, the source node keeps the message in its buffer and does not give copies to any forwarder nodes, but rather seeks the opportunity to encounter the destination node in order to deliver the message. This algorithm incurs the least cost as all the message forwards are actually message delivery actions. However, this algorithm does not guarantee a high delivery ratio or an adequate delay.

2) The **Epidemic** algorithm [32] is a social-and-power-oblivious opportunistic forwarding algorithm that broadcasts the message to the neighboring nodes. The epidemic spread of the message among the encountered nodes reduces the delay and maximizes the delivery ratio, at the maximum expense. Epidemic is considered a benchmark in opportunistic forwarding.

3) **PeopleRank** is a power-oblivious, social-aware message forwarding algorithm that forwards messages by utilizing the socially popular persons' nodes in place [14]. To achieve a balance between social-based forwarding and opportunistic forwarding, PeopleRank introduces a damping factor that decides the percentage of reliance on the social rank versus the opportunistic encounter of carriers. A damping factor (d) ranges from value 0 for total reliance on opportunistic forwarding, to 1 for total reliance on social-based forwarding.

The simulator implements the contact-aware PeopleRank version (CA-PeR) that ranks nodes using their social rank and social activeness. Node activeness is measured by how frequent a node encounters its social contacts. In effect, a node's social rank is either rewarded or penalized by the count of the node's encounters with its contacts. The simulator enables integrating interest and power awareness to the PeopleRank algorithm for simulating the IPeR [33] algorithm and the PIPeR algorithm versions [34].

4) **ProfileCast** [13] is a power-oblivious, social-aware paradigm that provides a service that delivers the message to all recipients who match a specified profile. It considers the history of mobility as the behavior profile (BP) in ranking carriers; similarity among these profiles directs the message forwarding process towards the targets who are identified by their profiles. SAROS implements the **CSI:D** protocol [13] of the ProfileCast paradigm. Based on a similar mobility pattern, CSI:D selects forwarders that share a common interest with the sender, while the target node's interest is orthogonal to that. Thus, message holders select the next proper message holder that satisfies these conditions: **1)** Any message holder prevents other nodes with similarity in BP within the neighboring threshold th_{nbr} from becoming message holders; **2)** Each message holder sends to nodes dissimilar to all known holders since their similarity in BP is less than the forwarding threshold th_{fwd}. SAROS requires both threshold values, and uses the similarity interest between the user and the forwarded message as the Behavior Profile.

5) **SocialCast** [12] is a power-oblivious interest-based routing protocol in DTNs supporting the publish-subscribe mechanism. It uses a prediction-based mechanism for guidance in the message holder selection, combined with store-and-forward to cope with intermittently connected networks. It relies on observing previous co-location and mobility patterns to predict the next mobility patterns of the users using Kalman filter forecasting techniques [35] for selection of the best message forwarders to reach the interested subscribers. SocialCast assumes that users with common interests tend to meet with each other more often than with other users. Thus, with high probability, they have similar mobility patterns and have high values of co-location with other nodes. The co-location and change in degree of connectivity attributes are included in the calculation of the utility value of each node which is used in the process of selecting the next message holders. This utility value is computed as follows:

$$Util(i) = w_{col} * P_{col}(i) + w_{cdc} * P_{cdc}(i) \qquad (1)$$

where $P_{col}(i)$ is the predicted co-location of node i with any of the target nodes. $P_{cdc}(i)$ is the predicted Change in Degree Of Connectivity of node i. The weights w_{col} and w_{cdc} respectively represent the relative importance of the co-location attribute and the change in degree of connectivity attribute. SAROS implements SocialCast and its proposed interest and power aware versions - ISCast, PISCast, and PISCastOp [36].

6) **SCAR** is a sensor context-aware routing protocol for opportunistic routing [37]. SCAR is a power-aware version of SocialCast. The SCAR protocol is applicable in opportunistic routing as it relies on existing routes between sensor nodes. It relies on forecasted values of the node's utility attributes based on the current values to improve performance. Its forecast

mechanism is based on observing the node's history of co-location with the destination nodes, change of degree of connectivity with other sensor nodes, and its power level. The utility function weighs all these attributes so that it maximizes the benefit gained from them and balances the trade-off among all these attributes. SAROS implements an adaptive SCAR version [37] which includes a monotonically decreasing function that adapts to the predefined ranges of values of a certain attribute. This version mainly sets an adaptive range for the battery level which monotonically decreases as the battery level enters a critical range. The predicted attributes are forecasted by applying Kalman Filter prediction techniques. SAROS also implements the proposed interest and power aware SCAR versions - ISCAR, PISCAR, and PISCAROp [36].

7) The **E-BubbleRap** [38] algorithm is the energy-aware version of the community-aware BubbleRap [25] routing algorithm. This energy-aware BubbleRap version combines socially-aware routing with energy consumption optimization [38]. However, this algorithm does not incorporate awareness of the forwarder node's interest in the forwarded content. E-BubbleRap introduces an energy-aware utility function of the original BubbleRap algorithm's local and global ranking functions which rank the nodes within their local communities and within the global community respectively.

8) The **Interest-Power Threshold Opportunistic Algorithm** is the interest-and-power-aware version of the Epidemic algorithm. Its main logic states that any message holder forwards messages to any neighboring node(s) whose similarity interest with the message $SInt(node, msg)$ is over a predefined interest threshold thr_{int}, and whose battery level $Bat(node)$ is above a predefined power threshold thr_{bat}. The forwarder selection condition is:

$$SInt(node, msg) \geq thr_{int} \;\;\&\;\; Bat(node) \geq thr_{bat} \quad (2)$$

E. Evaluation Module

This module evaluates the performance of the implemented algorithms in forwarding the message within the defined simulation period to meet a set of metrics categorized under the following three main categories: **1) Effectiveness:** ratio of interested forwarders, ratio of uninterested forwarders, recall, precision, accuracy, F-measure, and the Effectiveness Performance Index - the harmonic mean of F-measure, ratio of interested forwarders and ratio of uninterested forwarders; **2) Efficiency:** delivery ratio, cost represented in the number of forwarded copies of the message, and delay in delivering the message, and the Efficiency Performance Index - the harmonic mean of cost, delivery ratio and delay; **3) Power-awareness:** the ratio of the consumed power, the fairness index as defined in [39], the final battery distribution, the mean, standard deviation and variance among the final battery levels, and the Power-awareness Performance Index - the harmonic mean of fairness index and ratio of power consumption.

IV. The Simulator Interface and Usability

This section summarizes the interaction with the SAROS simulator in terms of a sophisticated interface, output graphs and exported data.

TABLE I. Parameters of the SAROS Simulator

No. of users	The number of users simulated in the experiment.
No. of iterations	The number of simulation runs in an experiment.
No. of shops	The number of advertisement senders. This number cannot exceed the no. of users
No. of interests	The length of the interest feature vector per user
Interest damping factor	The damping factor of PeopleRank and its interest and power aware versions.
Proximity Range	The range of detecting nodes in proximity in WiFi or Bluetooth modes. It is measured in meters
Source SInterest	The similarity interest of the sender with the sent message. It ranges from 0 to 1
Battery threshold	The battery threshold above which the node is considered a forwarding candidate.
dataset	Selects the SLAW or one of the real traces datasets
wCOL	The weight of the colocation component in the SocialCast and SCAR algorithms
wCDC	The weight of the change degree of connectivity component in the SocialCast and SCAR algorithms
Random Walk	Calls RandomSim to run random walk mobility based experiments
Batch mode	Runs in the experiments in batch mode
Start Sim	Runs the main code of the program that calls the selected algorithms
Full batteries	Sets the battery distribution to be full batteries
Normal Dist. batteries	Sets the battery distribution to the discrete normal distribution
PowerAware	Runs the power aware version of an algorithm
Opportunistic	Runs the opportunistic version of an algorithm
ContactAware	Runs the contact-duration aware version of PIPeR
Kalman Filter	Uses Kalman filter for contact-duration prediction
InterstAware	Runs the interest-aware version of an algorithm
Forwarding Algorithm	Runs the selected algorithms from this list: Epidemic, PeopleRank, IPeR, ProfileCast, BubbleRap, E-BubbleRap, SCAR and SocialCast
interest choice	Select the interest distribution.
destination percent	Sets the percent of the destination nodes in the community.
interested forwarder percent	Sets the percent of the interested forwarder nodes in the community.

A. The SAROS Interface

The main programs of the simulator and their main functionalities are: ***1) MainForm*** activates all the following forms. ***2) StartSim*** runs SLAW mobility-based experiments. It also can run the SIGCOMM/INFOCOM06 experiments. ***3) INFOCOMSim*** runs experiments with the imported and manipulated INFOCOM06 dataset. ***4) SIGCOMMSim*** runs experiments with the imported and manipulated SIGCOMM09 dataset. ***5) MallTracesSim*** runs experiments with an imported Mall-Environment encounter traces. ***6) RandomSim*** runs experiments using random walk mobility. ***7) StAndrewsSim*** runs experiments with the imported and manipulated St. Andrews University dataset mobility traces. The main parameters of the simulator are listed in Table I.

B. Output Graphs and Exported Data

The simulator outputs results in the form of graphs, metric files, and data files. The graph generation commands are produced by the SAROS simulator to generate the graphs via GNUPlot. The generated graphs are: **1)** The final battery distribution categorized into 4 categories; **2)** The battery distribution over time categorized into 4 categories as shown in Figure 2; **3)** The total consumed power over time; **4)** The total consumed power versus the delivery ratio; **5)** The count of produced control messages over time; **6)** Cost versus Delivery Ratio as shown in Figure 3; **7)** Cost over time; **8)** Delivery Ratio over time; **9)** Fairness Index; **10)** Recall, Precision, F-measure and Accuracy; **11)** The final delivery ratio, the percent of contacted interested forwarders, and the percent of

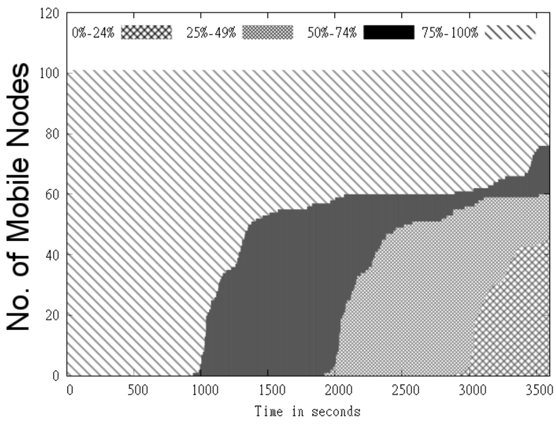

Fig. 2. Battery Distribution over Time

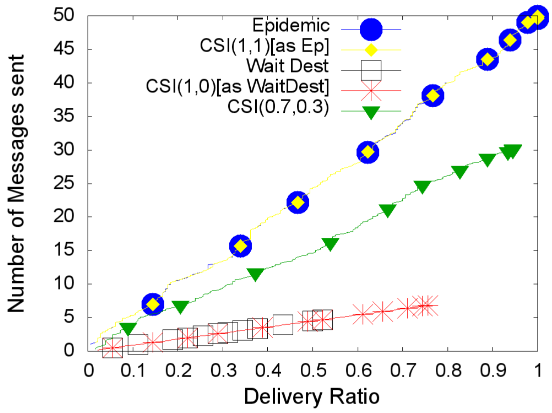

Fig. 3. Cost versus Delivery Ratio

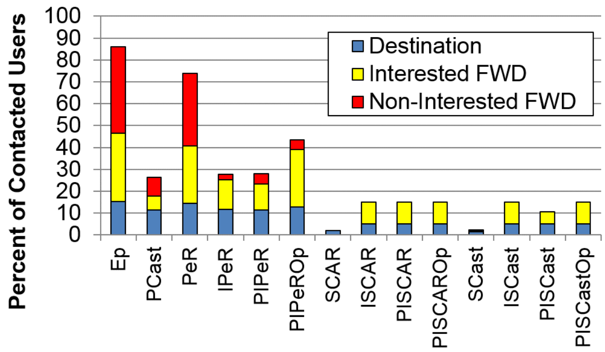

Fig. 4. Interest-based Effectiveness

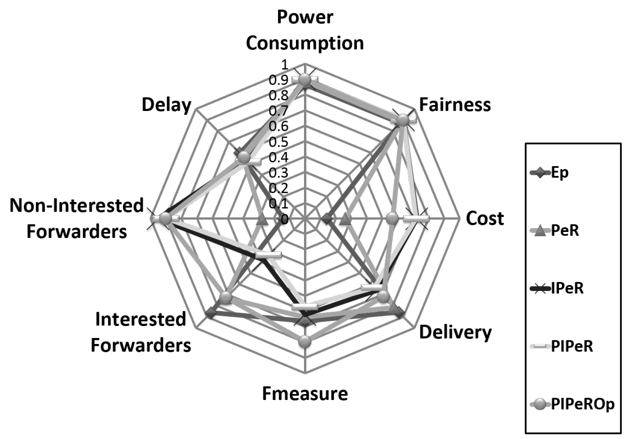

Fig. 5. The 8-Metric Analysis

contacted uninterested users as shown in Figure 4; **12)** The final mean and standard deviation of the battery levels; **13)** Variance among the battery levels over time; **14)** The power consumed due to the sent control messages; **15)** An 8-metric Spider graph which encompasses delivery ratio, cost, power consumption, fairness, interested forwarders ratio, uninterested forwarders ratio, F-measure, and delay as shown in Figure 5; **16)** The Performance Effectiveness Index: the harmonic mean of F-measure, ratio of interested forwarders and ratio of uninterested forwarders; **17)** The Efficiency Performance Index: the harmonic mean of cost, delivery ratio and delay; and **18)** The Power-awareness Performance Index: the harmonic mean of fairness index and ratio of power consumption.

The generated files of measured data or computed metrics for all algorithms are: **1)** The achieved delivery ratio, cost and consumed power per time slot; **2)** The recall, precision and accuracy; **3)** The final battery level per user per iteration; **4)** The delay exerted at the end of the simulation; **5)** The count of Wifi scans, forwards, receiving message per time slot; **6)** The fairness index; **7)** The final delivery ratio, the percent of contacted interested forwarders, and the percent of contacted uninterested users at the end of the simulation. **8)** Count of the incomplete message transfers, the total wasted time and the total wasted power due to incomplete message transfers. There is also a listing of the incomplete transfer occurrences; **9)** The generated interest distribution per run; **10)** The generated user profiles per time slot; **11)** The mean, the standard deviation and the variance of the battery levels per time slot; **12)** The final battery distribution per algorithm categorized into 4 categories ranging from [0%-24%] to [75%-100%]; and **13)** A Gnuplot commands file to generate the required graphs.

V. THE SIMULATOR VERIFICATION AND VALIDATION

We have conducted several experiments to verify the simulator code, including the following: **1)** To verify the implementation of the algorithms, we change the parameters of the algorithms to make them behave similarly to the Epidemic forwarding algorithm. We also run the Epidemic algorithm in the same environment to be able to compare the algorithms' performance to that of the Epidemic algorithm. The same verification method is applied by implementing the Wait Destination algorithm and adjusting the other algorithms' parameters so that they behave in a similar way to this algorithm. Then, we compare the results to verify the implemented code as shown in Figure 3; **2)** To verify the implementation of the KiBaM model, we run a set of experiments with no idle timeslots for the battery to prevent it from entering the charge recovery phase then compare its performance to that of another set of experiments that implement the simple battery; **3)** To verify the method of calculating variance, we plotted the battery levels of all the users as time passes, then we compared this graph to the produced variance graph; **4)** To verify the simulation of the implemented state-of-the-art algorithms, we compared their performance to that mentioned in the algorithms' original publications. The conducted evaluation proved the resemblance in the performance of both implementations.

VI. Conclusion and Future Work

SAROS is a simulator environment implemented in Visual C# to simulate opportunistic forwarding in environments such as malls, conference centers, or university campuses. User mobility traces are either synthetically generated or imported from real mobility traces. The simulator also generates users' social graphs and imports social graphs from real datasets. To represent user interest in the forwarded content, the simulator either generates random interest vectors for all users, or imports external interest vectors from other datasets. SAROS provides stable simulated opportunistic networking environments, and encompasses various opportunistic forwarding algorithms implementations. It also provides a variety of performance evaluation metrics not offered in comparable simulators. SAROS has been used in previous research contributions to validate our proposed interest and power aware opportunistic forwarding algorithms [33] [34] and to evaluate our proposed framework for integrating interest and power awareness in any social-aware opportunistic forwarding algorithm [36]. In the future, we intend to make SAROS available for public access to gather feedaback from other researchers, and to enable other enhancement and contribution from the research community.

References

[1] "Internet Live Stats," http://www.internetlivestats.com/internet-users/.

[2] "Facebook statistics," https://newsroom.fb.com/company-info/.

[3] "Youtube-statistics," http://www.youtube.com/yt/press/statistics.html.

[4] R. Kokku *et al.*, "Opportunistic Alignment of Ad Delivery with Cellular Basestation Overloads," in *MobiSys Proceedings*, 2011, p. 267.

[5] A. Mtibaa and K. Harras, "Exploiting Space Syntax for Deployable Mobile Opportunistic Networking," in *IEEE MASS*, 2013, pp. 533–541.

[6] A. Mashhadi *et al.*, "Ad Hoc Networks Fair content dissemination in participatory DTNs," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1633–1645, 2012.

[7] Y. Zhu *et al.*, "A Survey of Social-based Routing in Delay Tolerant Networks Positive and Negative Social Effects," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 387–401, 2013.

[8] K. C. A. Khaled A. Harras and E. M. Belding, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks," in *International Federation for Information Processing (IFIP) Networking*, 2005.

[9] K. A. Harras and K. C. Almeroth, "Inter-Regional Messenger Scheduling in Delay Tolerant Mobile Networks," in *IEEE WoWMoM*, 2006.

[10] C.-M. Huang, K.-c. Lan, and C.-Z. Tsai, "A Survey of Opportunistic Networks," in *22nd IEEE AINA Workshops*, 2008, pp. 1672–1677.

[11] A. Mtibaa and K. Harras, "CAF: Community Aware Framework for Large Scale Mobile Opportunistic Networks," *Computer Communications*, pp. 180–190, Jan. 2013.

[12] P. Costa *et al.*, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 748–760, Jun. 2008.

[13] W.-j. Hsu *et al.*, "CSI: A paradigm for behavior-oriented profile-cast services in mobile networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1586–1602, 2012.

[14] A. Mtibaa *et al.*, "Social Forwarding in Mobile Opportunistic Networks: A Case of PeopleRank," in *Communication and Social Networks*. Springer, 2012, vol. 58, pp. 387–425.

[15] A. Mtibaa and K. Harras, "FOG: Fairness in Mobile Opportunistic Networking," in *IEEE SECON*, 2011.

[16] O. R. Helgason and K. V. Jónsson, "Opportunistic Networking in OMNeT++," in *Proceedings of SIMUTools*, 2008, pp. 82:1–82:8.

[17] A. Keränen *et al.*, "The ONE Simulator for DTN Protocol Evaluation," in *Proceedings of SIMUTools*, 2009, pp. 55:1–55:10.

[18] S. Mehta *et al.*, "Network and system simulation tools for next generation networks: A case study," in *Modelling, Simulation and Identification*. InTech, August 2010, pp. 81–100.

[19] I. Rhee *et al.*, "CRAWDAD dataset ncsu/mobilitymodels (2009-07-23)."

[20] A.-K. Pietilainen and C. Diot, "CRAWDAD dataset thlab/sigcomm2009 (2012-07-15)."

[21] J. Scott *et al.*, "CRAWDAD trace cambridge/haggle/imote/infocom2006 (2009-05-29)."

[22] R. Friedman and A. Kogan, "On Power and Throughput Tradeoffs of WiFi and Bluetooth in Smartphones," in *IEEE INFOCOM Proceedings*, 2011, pp. 900 – 908.

[23] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," in *Proceedings of USENIX*, 2010.

[24] V. Rao *et al.*, "Battery model for embedded systems," in *18th International Conference on VLSI Design*, 2005, pp. 105–110.

[25] P. Hui *et al.*, "Bubble Rap: Social-based forwarding in Delay Tolerant Networks," in *9th ACM MobiHoc Proceedings*, 2008, pp. 241–250.

[26] "OPNET-Riverbed," http://www.riverbed.com/products/performance-management-control/opnet.html.

[27] M. S. Siraj *et al.*, "Network simulation tools survey," *IJARCCE*, vol. 1, no. 4, pp. 201–210, 2012.

[28] "OPNET-Riverbed IEEE 802.15.4/ZigBee OPNET Simulation Model," http://www.open-zb.net/wpan_simulator.php.

[29] N. Vallina-Rodriguez *et al.*, "Exhausting battery statistics understanding the energy demands on mobile handsets," in *ACM SIGCOMM MobiHeld workshop Proceedings*, 2010, pp. 9–14.

[30] C. G. Adriano Galati, "Human Mobility in Shopping Mall Environments," in *ACM MobiOpp Workshop Proceedings*, 2010.

[31] F. B. Abdesslem *et al.*, "CRAWDAD data set st_andrews/locshare (v. 2011-10-12)."

[32] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," CS-200006, Duke University, Tech. Rep., 2000.

[33] S. Al Ayyat, K. A. Harras, and S. G. Aly, "Interest Aware PeopleRank: Towards Effective Social-Based Opportunistic Advertising," in *IEEE WCNC*, 2013, pp. 4428–4433.

[34] S. Al Ayyat, S. G. Aly, and K. A. Harras, "PIPeR: Impact of Power-Awareness on Social-Based Opportunistic Advertising," in *IEEE WCNC*, April 2014.

[35] C. Chatfield, *The Analysis of Time Series: An Introduction*, 6th ed. Chapman and Hall/CRC, 2004.

[36] S. Al Ayyat, K. A. Harras, and S. G. Aly, "On the Integration of Interest and Power Awareness in Social-Aware Opportunistic Forwarding Algorithms," *Computer Communications*, pp. 97–110, Nov. 2015.

[37] B. Pasztor *et al.*, "Opportunistic mobile sensor data collection with SCAR," in *IEEE MASS*, 2007, pp. 1–12.

[38] C. Chilipirea, A.-C. Petre, and C. Dobre, "Energy-aware social-based routing in opportunistic networks," *Int. J. of Grid and Utility Computing*, vol. 1, no. 3/4, p. 1, 2013.

[39] A. J. Mashhadi *et al.*, "Fair content dissemination in participatory DTNs," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1633–1645, 2012.