The Seventh IEEE Workshop on Pervasive Collaboration and Social Networking, 2016

Friend-to-Friend Short Message Service with Opportunistic Wi-Fi Beacons

Okan Turkes, Hans Scholten, and Paul J. M. Havinga
Pervasive Systems, EEMCS, University of Twente, 7500EA, Enschede, NL
E-Mails:{o.turkes,j.scholten,p.j.m.havinga}@utwente.nl

***Abstract*—This study introduces Oppline, an ad hoc opportunistic short message service that can be collaboratively used by everyone who has a smart mobile device. The data exchange method of Oppline is built on top of the universal Wi-Fi standard, thus expedites platform-independent integration of related mobile applications for delay-tolerant communications in public space. Even in highly-dense mobile networks of smart portable devices, Oppline gains performance from people's participation according to our experimental analysis validated with a real-life deployment. Without creating any network overhead, Oppline performs multi-hop message transmissions via short messages encoded in and decoded from Wi-Fi's service set identifier field. As a ubiquitous alternative to the situated communication systems, Oppline provides high versatility and usability for mobile ad hoc applications based on friend-to-friend messaging and data dissemination.**

I. INTRODUCTION

This work describes the design, implementation, and experimental analysis of Oppline, a smart mobile ad hoc networking service for short messaging between people at overcrowded events such as concerts, festivals, sports competitions, and more. During such events, people can get separated from their families or group of friends without any chance to regather. Besides, people are often confronted with internet access issues to reach their families or friends with their mobile handsets. From local networks (Bluetooth, Wi-Fi) to cellular short message service (SMS) to broadband GSM (3G, 4G), situated communication systems in crowds represent critical connectivity limits, and might sometimes be overloaded due to huge uplink/downlink demand from high number of mobile devices such as smartphones and tablets [1], [2]. In contrast, Oppline benefits from a crowd of smart mobile devices to deliver short messages to and from group of people by using opportunistic Wi-Fi Service Set Identifier (SSID) broadcasts. For hopping of messages towards their destinations, it alternately functions SSID beaconing and scanning with the use of Wi-Fi Hotspot and Wi-Fi Infrastructure modes, respectively. Without IP layer connection, messages are encoded in SSID fields, are announced in Wi-Fi Hotspot mode, thus are directly delivered to the scanning devices in proximity. Devices employ a continuous beacon/scan switching to provide multi-packet multi-hop transmissions.

As illustrated in Figure 1, Oppline is suitable for use on any kind of device supporting SSID beaconing, in particular smartphones, tablets, smartwatches that people always carry. It can run on top of any affiliated Wi-Fi adapter without any modification or installments on IEEE 802.11 standards. Moreover, it connects the diversity of smart mobile devices running different platforms such as Android, iOS, and Windows Phone. It is a bi-directional multi-hop networking model which does not get affected by network overhead. As an association-free protocol, it uses the wireless broadcast advantage. As Figure 1 delineates, multiple beacons can be received by a single scan operation and a beacon can be received by multiple scans at a single time. Thus, Oppline manages the distribution of multiple request and response messages between and through people without establishing connections. Even in dense networks, it facilitates ”friend finder” type opportunistic SMS and other daily-life SMS types as well as data dissemination services.

Oppline is evaluated with a comprehensive set of simulated network scenarios. The simulations are validated with the real-world experiments conducted with 20 Android smartphones. The feasibility of Oppline is investigated in terms of the energy efficiency and capability of its Wi-Fi operations. From small-scale to large-scale, from sparse to dense, the unicast routing performance of Oppline is assessed through various network setups. The same setups are assessed for the multicast routing and broadcast routing efficiency as well. In each setup, different message creation intervals ranging between 1 minute to 15 minutes are studied with different number of devices ranging from 50 up to 200. According to our performance analysis, Oppline gains advantage from ubiquity of smart mobile devices to provide a reasonable end-to-end connectivity in dense network environments.

The rest of the paper is organized as follows: Section II discusses the related works. Section III presents the communication model. Section IV describes the implementations for experimental setup and elaborates on the performance analysis. Section V gives an overall discussion on Oppline. Finally, Section VI concludes the paper with the future works.

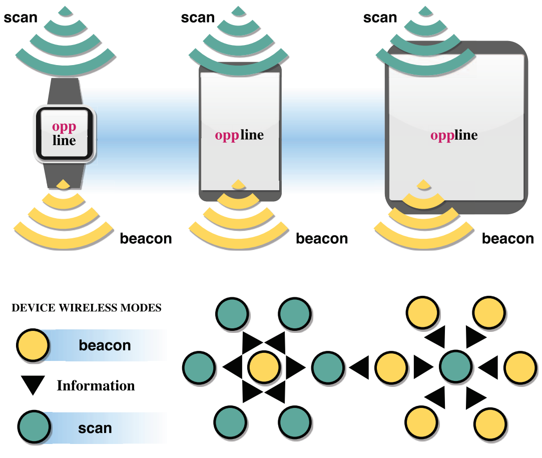


Fig. 1. Oppline's data forwarding model

978-1-5090-1941-0/16/$31.00 ©2016 IEEE

II. Related Works

For friend-to-friend opportunistic short messaging, a large variety of mobile applications and services are proposed for public end-use. The early examples for decentralized messaging involve simplistic single-hop solutions. Nokia Sensor [3] enables users to detect others in the vicinity via classic Bluetooth. Similarly, Beacon Friend Finder [4] also uses classic Bluetooth signals to send out identifiable user messages. The problem of classic Bluetooth device discovery is the frequent and time-consuming user intervention for device pairing. Eliminating the manual pairing process, E-Smalltalker [5] proposes encoding of Bluetooth device names to carry user-defined contents. Apart from Bluetooth, E-Shadow [6] and Help Beacons [7] extend the P2P messaging in social spaces via user contents advertised in 802.11 SSID fields. In contrast to these studies, Oppline employs a data exchange protocol which is multi-hop, bi-directional, and highly opportunistic.

Decentralized messaging systems in crowds necessitate an assembly of multi-hop, bi-directional, and opportunistic protocols. The current wireless P2P and ad hoc standards on mobile platforms, i.e. Bluetooth and Wi-Fi support multi-hop bi-directional interfaces. However, these interfaces have certain limitations such as publicly-limited functions requiring root access privileges [8] and issues under frequent disconnections caused by high device density and mobility [2]. A number of studies exist supporting opportunistic communications for general-public use. In [8], WiFi-Opp provides delay-tolerant ad hoc network support on smartphones with a constant switch between Wi-Fi Hotspot and Wi-Fi Infrastructure modes, that is quite similar to Oppline's duty cycling mechanism. FireChat [9] presents a mesh networking application on smartphones with the use of either Wi-Fi multi-peer connectivity interface or Bluetooth Low Energy peripheral mode. As connection-based methods, both WiFi-Opp and FireChat provide self-organizing ad hoc networks on smart mobile devices, nevertheless their networking models are i) bound to high overhead of network discovery in mobile environments, and ii) small-scale examples due to the limitations on device connection numbers for both Wi-Fi and Bluetooth in mobile operating systems [10]. In contrast, Oppline is a more opportunistic method supporting spontaneous data sharing without creating any communication overhead. Moreover, Oppline can be readily integrated to any group and type of smart mobile devices. Ignoring connections, Oppline completes data delivery at the neighbor discovery stage, and thus provides high flexibility of data sharing even in highly-dense and highly-mobile networks. To the best of our knowledge, Oppline is the first general-public smart mobile application supporting decentralized opportunistic multi-hop short messaging in crowded places.

III. Communication Model

Our solution to the problem of decentralized opportunistic short message communications with smart mobile devices is based on a specific use-case of the *opportunistic beacon networking* (OBN) model presented in [11]. In brief, OBN is a highly opportunistic data switching model for smart mobile devices working through wireless network identifiers. Oppline operates OBN on top of the Wi-Fi standard to provide data switching with encoding of SSID fields. For brevity and clarity, Table I lists a set of notations used in the rest of the paper.

TABLE I. Notations

Symbol	**Definition**
d_i	i-th device in a network
N	Total number of devices in a network
M_{REQ}	Total number of created requests in a network
M_{ACK}	Total number of created acknowledgments in a network
M	Total number of created messages in a network, $M_{REQ} + M_{ACK}$
t_{OB}	Duration of an OB state
t_{BO}	Duration of a BO state
t_{XOB}	Transition duration from BO to U&S to OB
t_{XBO}	Transition duration from OB to U&S to BO
T	Period of an automaton cycle, i.e. $t_{OB} + t_{XBO} + t_{BO} + t_{XOB}$
t_{SI}	Scan interval
t_{BI}	Beacon interval

For d_i's having at least one message, Oppline employs a finite state automaton with 3 states as Figure 2 illustrates: *Opportunistic Beacon* (OB): Broadcast of encoded SSID fields in Wi-Fi Hotspot mode, *Beacon Observer* (BO): Scanning of SSID fields within proximity in Wi-Fi Infrastructure mode, and *Update & Switch* (U&S): transition between OB and BO.

Oppline has a continuous duty cycling between the OB and BO states in order to provide two-directional data exchange. U&S is an idle state to switch between beaconing and scanning functionalities, which is imperative for today's Wi-Fi adapters. At BO state, discovered SSIDs (messages), if any, are decoded. The received (and locally created) messages are stored in a circular queue, Q. To sustain multi-packet transmissions, the messages in Q are selected for SSID encoding in a circular order in advance of each OB state. At each OB state, therefore, a different message is broadcast in the SSID field. If there is only one message in Q, OB state repeatedly broadcasts that one. If Q is empty, the automaton stands by at BO mode until a message is discovered. The size of Q can be determined based on network type or application needs. Since large queueing can cause message starvation, the size of Q can be kept fixed, allowing the newest messages to overwrite the oldest ones.

Our design comprises two message types: *Request* (`REQ`) and *Response* (`ACK`). As Figure 3 demonstrates, any d_i in a given network can send out a `REQ` towards a determined destination d_j through other devices. Once d_j receives the

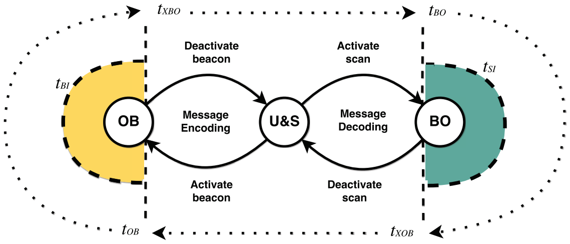


Fig. 2. Data exchange automaton

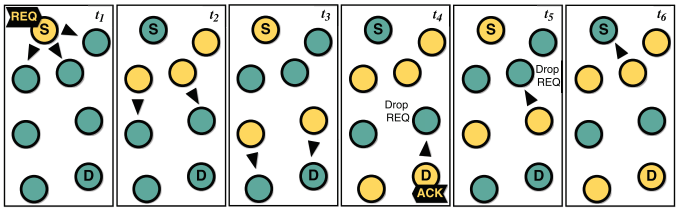


Fig. 3. Data routing example

`REQ`, it immediately creates an `ACK` back to d_i. Relay devices in between d_i and d_j delete from their Q the REQ of a discovered ACK since it is not necessary anymore. This also increases the probability of ACK relays in the duty cycles. Based on the contact opportunities and duty cycles of the devices in range of each other, the number of hops and paths can vary for each message created in the network.

A. *Model Parameters*

Each state of the automaton has specific service durations. The durations of OB and BO are t_{OB} and t_{BO}, respectively, which can be adjusted based on the network or application needs. At BO state, *scan* operation repeats in every t_{SI}. At OB state, *beacon* operation repeats transmitting of a packet in every t_{BI}. At U&S state, *activate* and *deactivate* operations together defines the switching duration. Transition from OB to BO, i.e. deactivating Wi-Fi Hotspot mode and activating Wi-Fi Infrastructure mode has a total duration of t_{XBO}. Transition from BO to OB, i.e. deactivating Wi-Fi Infrastructure mode and activating Wi-Fi Hotspot mode takes t_{XOB} in total.

Running the automaton with a random initialization time, a d_i can be in either of three states at a particular time. In our design, $t_{OB}{=}t_{BO}$ for all d_i's in order to provide a fairness factor between the *beacon* and *scan* operations. If a group of d_i's in proximity of each other occurs at the same state with concurrent transitions, the data exchange will never work between them. Therefore, a randomization for T for each d_i is required to reduce the possibility of OB-OB and BO-BO conflicts which may abide forever. The randomization of T is provided with non-deterministic t_{XOB} and t_{XBO} values which are investigated with real-world experiments in Section IV.

B. *Message Encoding*

A default SSID field can contain at most 32 ASCII bytes. Based on the application requirements, Oppline messages can be designed in various compositions of routing-specific data. Figure 4 shows the SSID encoding types of our design over two examples: i) a unicast message for end-to-end routing, and ii) a broadcast message for data dissemination. In order to encode values of several fields in less number of ASCII bytes, `Base94` conversion is applied since an SSID byte can have 94 different ASCII characters. In our design, a unicast message encoding consists of 6 fields:

1) Preamble (2 bytes): A distinctive tag to help BOs distinguish the message in an application or a network.
2) Message creation time (5 bytes): UNIX time is used in our design. However, shorter time formats such as `HHmm` converted to `Base94` can also be used for compactness in exchange for less timing precision.
3) `REQ`/`ACK` type (1 byte): A predefined type of a request or a response that represents a particular message or context.
4) Location information (8 bytes): The position of or information about the location where the message is created. GPS is used in our design. For indoor applications, the location can be reported by user input.
5) Source Device Identifier (8 bytes): MAC address is used in our design. An alternative such as device ID, username, and so on can also be used instead.
6) Destination Device Identifier (8 bytes). Same as (5).

A broadcast message encoding, on the other hand, may discard routing-related fields such as device identifiers to gain length for the message field to be publicly disseminated.

IV. IMPLEMENTATION & EVALUATION

This section gives the implementations, experiments, model evaluation parameters and metrics, and performance analysis.

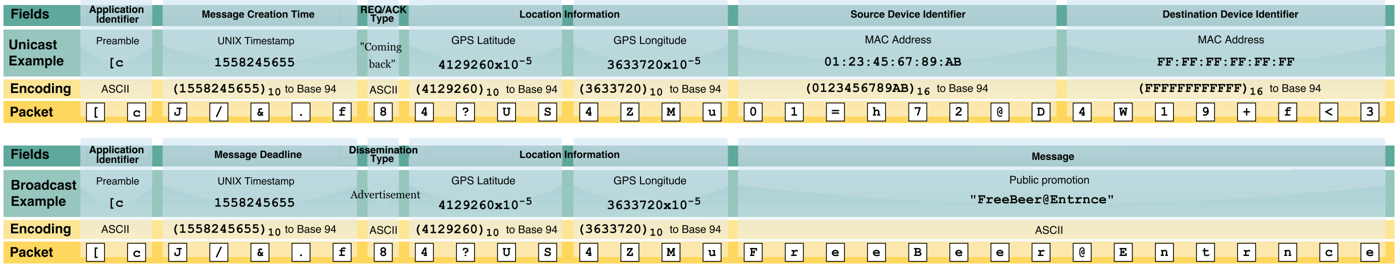
A. *Mobile Application*

Oppline is implemented as an Android application for testing purposes. The application employs the OBN protocol [12] with the utilization of Wi-Fi Hotspot and Wi-Fi Infrastructure modes in an alternating manner for the OB and BO roles, respectively. As defined in the protocol, Wi-Fi Inrastructure mode constantly runs unless a message is created or scanned. If a message is available, SSID is encoded to that message in advance of beaconing with Wi-Fi Hotspot mode. In each duty cycle, the application selects the front-most packet of Q.

The mobile application tests are held with various networks formed of Samsung S4 Mini and Motorola Moto G phones. A total of 20 smartphones are utilized in order to collect time measurements of Wi-Fi operations used in the model as well as to investigate Oppline's networking performance in reality. The measurements are further used in the simulation modelling. On the other hand, the networking results are compared with that of the simulation runs in order to verify the simulator.

B. *Simulator*

In addition to the mobile application, a cycle-based simulator is implemented in *Matlab* that can run the Oppline's data exchange protocol for any kind of network setup. The simulator creates messages in a given network setup as discrete events. At a particular instant in time, each message is created within a simulated device. Devices mimic the data exchange



Fields	Application Identifier	Message Creation Time	REQ/ACK Type	Location Information		Source Device Identifier	Destination Device Identifier
Unicast Example	Preamble [c	UNIX Timestamp 1558245655	"Coming back"	GPS Latitude $4129260\text{x}10^{-5}$	GPS Longitude $3633720\text{x}10^{-5}$	MAC Address 01:23:45:67:89:AB	MAC Address FF:FF:FF:FF:FF:FF
Encoding	ASCII	$(1558245655)_{10}$ to Base 94	ASCII	$(4129260)_{10}$ to Base 94	$(3633720)_{10}$ to Base 94	$(0123456789AB)_{16}$ to Base 94	$(FFFFFFFFFFFF)_{16}$ to Base 94
Packet	[c	J / & . f	8	4 ? U S	4 Z M u	0 1 = h 7 2 @ D	4 W 1 9 + f < 3

Fields	Application Identifier	Message Deadline	Dissemination Type	Location Information		Message
Broadcast Example	Preamble [c	UNIX Timestamp 1558245655	Advertisement	GPS Latitude $4129260\text{x}10^{-5}$	GPS Longitude $3633720\text{x}10^{-5}$	Public promotion "FreeBeer@Entrnce"
Encoding	ASCII	$(1558245655)_{10}$ to Base 94	ASCII	$(4129260)_{10}$ to Base 94	$(3633720)_{10}$ to Base 94	ASCII
Packet	[c	J / & . f	8	4 ? U S	4 Z M u	F r e e B e e r @ E n t r n c e

Fig. 4. Message Encoding

protocol with specified network parameters. The simulator is run with an abstract Wi-Fi PHY/MAC modelling. In reality, several Wi-Fi operations have uncontrollable execution times. The simulator is fed t_{XOB}, t_{XBO}, t_{BI}, and t_{SI} values collected from all of the mobile application runs. Table II demonstrates the average and standard deviation values of these values. Out of all real-world measurement values, simulated devices pick a random group of t_{XOB}, t_{XBO}, t_{BI}, and t_{SI} for each of their duty cycle.

C. Simulation Validation

In order to investigate its regularity regarding the data exchange protocol, the simulator is compared with the mobile application runs conducted in a real-world network deployment. Both real-world experiments and simulations are run with the same varying model parameters in order to find a correlation between their performance outcomes. Real-world experiments consist of 20 phones that are situated in range of each other. The simulation is set up in the same manner with the same number of devices. All tests are performed under no mobility. Thus, it is aimed to discover the accuracy of the simulation model when mobility effect is discarded.

The real-world experiments and simulations are run 3 times and 500 times, respectively, per unique parameter combination. Figure 5 shows the collated results for data dissemination performance and end-to-end latency performance of the model. Under different message creation intervals (shown as t_{MI}) settings ranging from 30s up to 240s, the real-world test results and the simulation test results are shown together for two different $t_{OB} = t_{BO}$ settings, 25s and 45s. For all network settings, the simulation results are in line with the real-world test results. Nevertheless, minor deviations are notable especially for the latency results. These deviations might be related either to the limited number of the physical world test runs, or to the inimitable environmental factors affecting the PHY/MAC operations in reality. Nonetheless, it is evident that the simulation results highly correlate with the corresponding results obtained from real-world experiments. The simulator is used for the evaluation of our test setups.

D. Test Setups & Model Evaluation Parameters

All tests are conducted by means of simulations with a set of controlled experiments based on varying network types and model evaluation parameters. In practice, our aim is two-fold:

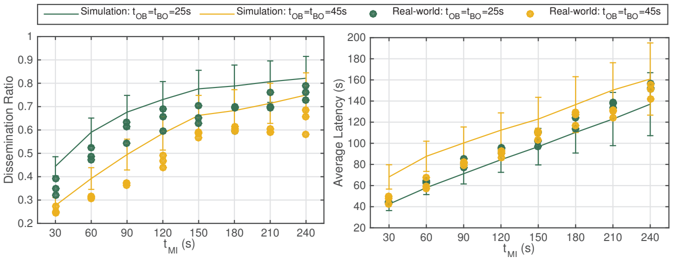


Fig. 5. Simulation validation test results grouped under different t_{MI} settings. For each t_{MI}, the real-world results are shown as a collection of 3 runs. For the simulations, the averaged results of 500 runs are represented. The end of the error bars indicate the standard deviation values. In all tests, $|Q|$=10.

TABLE II. REAL-WORLD DURATIONS USED IN THE SIMULATOR

Symbol	Duration	Symbol	Duration
t_{BI}	μ =0.100s, σ =0.014s	t_{SI}	μ =3.000s, σ =0.247s
t_{XOB}	μ =4.302s, σ =0.524s	t_{XBO}	μ =3.407s, σ =0.327s

μ shows the average value, σ shows the standard deviation value

TABLE III. SIMULATION PARAMETERS

Duration: 14400s, Repeats: 100	
Number of devices: $N_1 = 50, N_2 = 100, N_3 = 150, N_4 = 200$	
Networks $(m \times m)$: $S_1 = 500 \times 500$, $S_2 = 750 \times 750$, $S_3 = 1000 \times 1000$	
Mobility: Random Waypoint	Movement pause (s): $[0,1800]$
Wi-Fi range (m): $50 \pm [0, 25]$	BLE range (m): $40 \pm [0, 20]$
$t_{OB} = t_{BO} = 15s$	$t_{BI}, t_{SI}, t_{XOB}, t_{XBO}$: Table II
$t_{MI}(s) = \{60, 180, 360, 540, 720, 900\}$, $\lvert Q \rvert = 10$	

i) to assess Oppline's performance under varying network densities. 4 different network groups are formed: N_1, N_2, N_3, and N_4 consist of 50, 100, 150, and 200 devices, respectively. Besides, 3 different deployments are formed with the following network sizes: S_1: $500m \times 500m$, S_2: $750m \times 750m$, and S_3: $1000m \times 1000m$. The tests are taken for all possible combinations between the network groups and deployments.

ii) to assess Oppline's performance under varying message numbers. 6 different values for message creation interval (t_{MI}) are tested, ranging from 60s up to 900s. The tests with different t_{MI} are taken for all combinations in (i).

Each network setup is repeated for 100 times. Table III shows the simulation parameters in brief. For the device movements, random waypoint mobility model with movement pauses is utilized. The devices are given random radio ranges uniformly ranging from 25m to 75m.

E. Evaluation Metrics

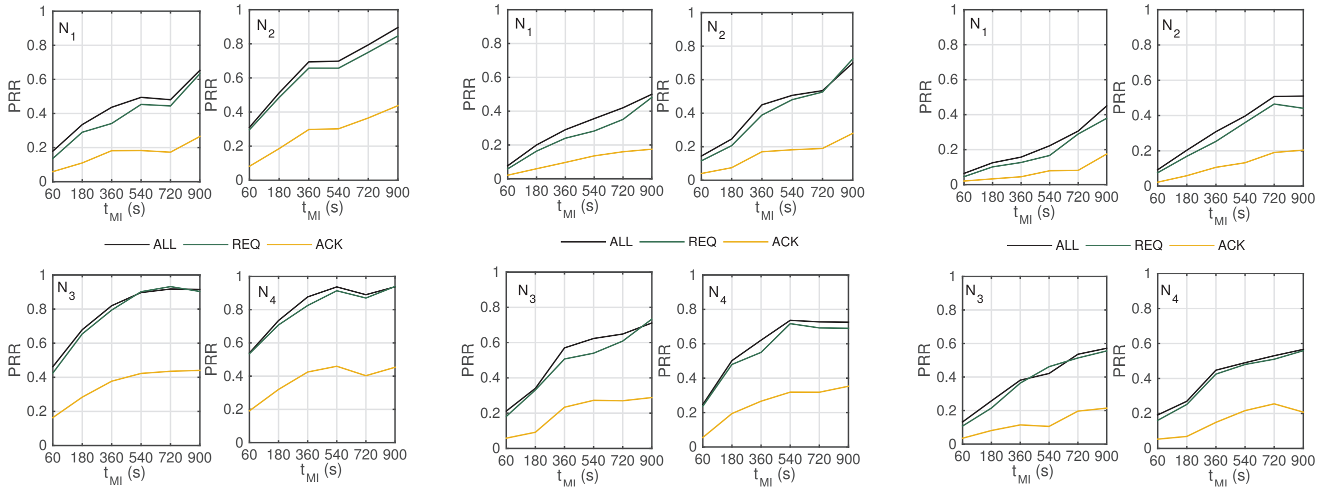
In a unique network run, let us define M^+ as the number of delivered messages out of M, $M^+ < M$. Same definition holds for $M_{REQ} \rightarrow M^+_{REQ}$ and $M_{ACK} \rightarrow M^+_{ACK}$. Similarly, let us define $M*$ as the total number of message copies out of M messages created in d_i's. The network setups are evaluated with the following metrics:

1) Message reception rate (PRR), calculated as $\frac{M^+}{M}$.
   The PRR of `REQ`s (PRR$_{REQ}$) is calculated as $\frac{M^+_{REQ}}{M_{REQ}}$.
   The PRR of `ACK`s (PRR$_{ACK}$) is calculated as $\frac{M^+_{ACK}}{M}$.
2) Latency (L), the delivery time of a message between its source and destination. L_{REQ} is the sum of the delivered `REQ` latencies, divided by M^+_{REQ}. L_{ACK} is the sum of the delivered `ACK` latencies, divided by M^+_{ACK}.
3) Average dissemination ratio (D), calculated as $\frac{M*}{M \times (N-1)}$.

F. Performance Analysis

The results of the experiments are presented in this section.

Figure 6 shows the unicast data delivery performance results grouped under S_1, S_2, and S_3. In each group, N_1, N_2, N_3, and N_4 are shown as separate plots. In each plot, the average PRR values for M, M_{REQ}, and M_{ACK} are shown as separate line graphs demonstrated on a t_{MI} scale.



(a) S_1 (b) S_2 (c) S_3

Fig. 6. Unicast data delivery performance

In Figure 6(a), the increase in the average PRR values is quite remarkable as t_{MI} increases. In other words, any deployment can handle data delivery more efficiently when the number of messages decreases in the network. When t_{MI}=60s, PRR$_M$ is 19% for N_1, gradually increases as N increases, and reaches to 57% for N_4. The difference in PRR$_M$ between t_{MI} =60s and t_{MI} =360s is ≈25% for N_1. This difference ranges between ≈30% and ≈40% for N_2, N_3, and N_4. These results simply indicate the positive effect of high device numbers on the delivery performance. When t_{MI} =900s, all N setups except N_1 attain a PRR$_M$ of more than ≈85%. For N_4, which is the densest setup of our experiments, a PRR$_M$ of 93% is achieved as the maximum. In Figure 6(b), in which the PRR values of the S_2 setups are depicted, a similar trend is present as in S_1. Nevertheless, since the network size is bigger, the decrease in the overall delivery performance for the same N setups is also remarkable. In comparison to PRR$_M$ values obtained in S_1, that of in S_2 are from ≈15% to 35% lower when t_{MI}=60s. Of PRR$_M$ on the scale of t_{MI} ranging between 360s and 900s, the difference in the overall performance between the S_1 and S_2 setups decreases to ≈20%. A correlative decrease in the overall delivery performance for the same networks deployed in S_3 is shown in Figure 6(c). Compared to the S_2 results, there exists an average of 12% drop in PRR$_M$ for all of the results obtained in S_3 setups.

In all tests, the PRR$_{ACK}$ values which demonstrate the average round trip delivery efficiency are roughly 50% lower than their corresponding PRR$_{REQ}$ values. This means that only approximately half of the sources in all setups can get a response back from their destinations. On the other hand, the PRR$_{REQ}$ values follow a high performance in parallel with the PRR$_M$ values in all tests. Note that M=M_{REQ}+M_{ACK} and M_{ACK}=M^+_{REQ}. In any setup, M is influential on PRR$_{REQ}$, and consequently PRR$_{REQ}$ is influential on PRR$_{ACK}$. Since `REQ`s of the discovered `ACK`s are dropped in the device buffers, L_{ACK} performs better than L_{REQ} and L_M as Figure 7 depicts for S_1. Overall, L_M significantly decreases as N and/or t_{MI} increases. For low t_{MI}, it is shown that a message is delivered in average of 600s for N >50. For t_{MI} >540s, the average latencies fall below 400s. For t_{MI} =900s, all latencies are around 180s. From the latency results, it is also possible to assess the round trip time (RTT) of the delivered `REQ`-`ACK` pairs. L_{REQ}+L_{ACK} gives the average RTT. In all network setups except N_1, the RTT ranges between 320s (for t_{MI}=900s) up to 780s (for t_{MI}=60s) in average.

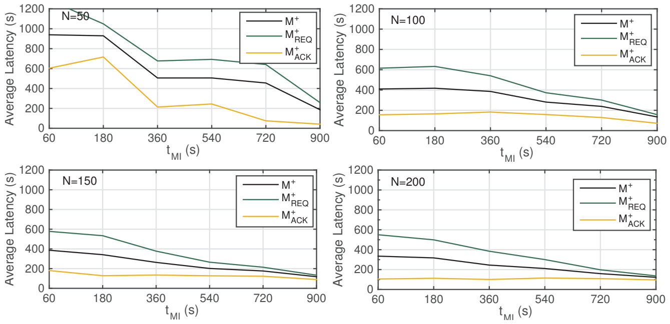


Fig. 7. Unicast latency performance under S_1

In addition, Figure 8 depicts how the average D_M and the average L_M changes over the simulation time for N_4 under S_3, i.e. to demonstrate Oppline's dissemination efficiency through high number of devices dispersed in a sparse network. For all of the demonstrated setups, both D_M and L_M succeed a steady performance during the simulation runs. The fixed length of Q, which is 10 in our experiments, has an influence on these persistent results obtained throughout the network operation. The reason is that, newly-created messages which overwrite the oldest ones are not affected by message starvation even in the scenarios having high M. For instance, when t_{MI}=60s, the network operation can still sustain a D_M of ≈43% at 14400s. Similarly, the L_M values reside between 300s and 400s. As the message creation interval increases, the dissemination performance increases. When t_{MI}=360s, a D_M of ≈80% can be achieved and a L_M of ≈270s can be provided. The D_M values obtained when t_{MI} =720s are between the range of ≈80% and ≈85%, but with lower L_M values around ≈180s in average.

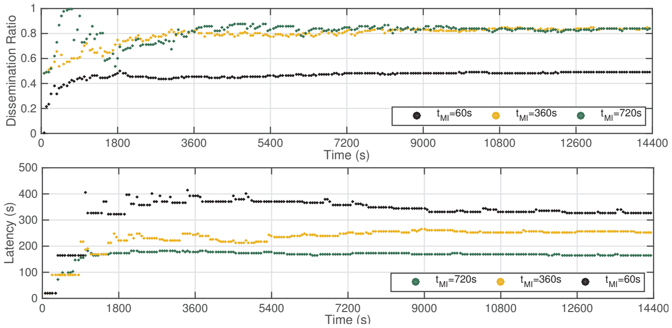


Fig. 8. Data dissemination and latency over time with N_4 under S_3

V. DISCUSSION

The presented results signify three prominent outcomes:

i) Oppline achieves a high message delivery performance in dense deployments. As the densest setup, N_4 under S_1 gives an average PRR of 93% when t_{MI}=900s. For the same network setup, the average PRR declines to 54% when t_{MI}=60s. In terms of delivery latency, Oppline also maintains a high performance in dense deployments. The average L values obtained in all tests fall in the range of 180s and 400s in average for N_2, N_3, N_4 under S_1.

ii) Oppline provides high suitability for low message frequency scenarios, or for the opportunistic ad hoc networks having single (or limited number of) message sources. Regardless of network density, the networking performance significantly increases as t_{MI} is increased. Having the results from all network setups, the average PRR ranges between 50% and 95% when t_{MI}=900s.

iii) Oppline can also be used in data dissemination scenarios. As the densest setup, N_4 under S_1 provides an average D_M of 74% when t_{MI} =60s and an average D_M of 91% t_{MI} =900s throughout the tests. These results are not included in the paper due to space constraints.

Overall, Oppline is a lightweight opportunistic ad hoc routing and dissemination service with certain restrictions in its protocol design. According to our PRR$_{ACK}$ results, the bi-directional routing of Oppline provides a moderate performance under reasonable network densities. Nonetheless, especially in crowded places, it can be a substantial alternative to the situated communication systems in the absence of an online network operation. PRR$_{REQ}$ results point out that Oppline can be an opportunistic line between group of friends as well as can be used to inform people in crowded places.

The design and implementation of Oppline is investigated with Wi-Fi beacons. However, the data exchange protocol of Oppline can be designed above any other wireless PHY/MAC protocol supporting beaconing. For instance, universal unique identifier (UUID) fields of Bluetooth Smart (Low Energy) can be exploited in our protocol as well. Suppporting any kind of wireless PHY/MAC, Oppline constitutes a good example for platform-independent ad hoc short messaging services.

VI. CONCLUSION & FUTURE RESEARCH DIRECTIONS

This paper has presented Oppline—an ad hoc opportunistic short messaging service which expedites lightweight and universal mobile opportunistic communications via wireless network identifiers. Oppline is intended for smart mobile devices to support decentralized bi-directional and multi-hop short messaging between peers in crowded events such as concerts and sports organizations. Oppline is based on an automatic data exchange protocol that can operate on top of Wi-Fi without requiring any modification or installment. Providing an ease of applicability, the protocol can therefore directly operate on any smart mobile platform. Oppline can be readily integrated to the mobile opportunistic messaging applications. Moreover, inexperienced end users can easily take part in such applications.

The performance evaluation of Oppline has been conducted over a large set of simulations validated with a real-world deployment. For the real-world tests, Oppline is developed as an *Android* and its networking is studied with 20 smartphones. The simulations are conducted to study Oppline under larger setups. Our results clearly indicate that Oppline achieves a promising delivery performance in point-to-point scenarios under reasonable device density.

Our future research directions include several improvements on the presented model. Our initial aim is to investigate several adaptive schemes for different Oppline use cases. Second, Oppline will be tested in several real-world deployments at such as, but not limited to, campus areas, crowded places, traffic environments.

ACKNOWLEDGMENTS

This paper is supported by the SenSafety project within the context of the Dutch National Program COMMIT.

REFERENCES

[1] M. Luna and R. Bott, "Managing cache to prevent overloading of a wireless network due to user activity."

[2] S. Sagari, A. Baid, I. Seskar, T. Murase, M. Oguchi, and D. Raychaudhuri, "Performance evaluation of mobile hotspots in densely deployed wlan environments," in *IEEE PIMRC 2013*.

[3] P. Persson and Y. Jung, "Nokia sensor: From research to product," in *Proceedings of the 2005 Conference on Designing for User eXperience*, ser. DUX '05, 2005.

[4] "Beacon Friend Finder," http://bff-app.com, accessed: 2015-11-19.

[5] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, 2010, pp. 468–477.

[6] J. Teng, B. Zhang, X. Li, X. Bai, and D. Xuan, "E-shadow: Lubricating social interaction using mobile phones," *Computers, IEEE Transactions on*, vol. 63, no. 6, pp. 1422–1433, 2014.

[7] A. Al-Akkad, L. Ramirez, A. Boden, D. Randall, and A. Zimmermann, "Help beacons: Design and evaluation of an ad-hoc lightweight s.o.s. system for smartphones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14, 2014.

[8] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre, "Wlan-opp: Ad-hoc-less opportunistic networking on smartphones," *Ad Hoc Networks*, vol. 25, Part B, pp. 346 – 358, 2015.

[9] "Fire Chat," http://opengarden.com, accessed: 2015-09-30.

[10] O. Turkes, H. Scholten, and P. J. Havinga, "Blessed with opportunistic beacons: A lightweight data dissemination model for smart mobile ad-hoc networks," in *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks*, ser. CHANTS '15, 2015, pp. 25–30.

[11] O. Turkes, H. Scholten, and P. J. M. Havinga, "Opportunistic beacon networks: Information dissemination via wireless network identifiers," in *review*.