PASA: Passive Broadcast for Smartphone Ad-hoc Networks

Ying Mao Jiayin Wang Joseph Paul Cohen Bo Sheng
University of Massachusetts Boston
{yingmao, jane, joecohen, shengbo}@cs.umb.edu

***Abstract*—Smartphones have become more and more popular in the past few years. Motivated by the fact that location plays an extremely important role in mobile applications, this paper develops an efficient local message dissemination system PASA based on a new communication model called *passive broadcast*. It is based on the method of overloading device names described in MDSRoB [14] and Bluejacking [23]. In this new model, each node does not maintain connection state and data delivery is initialized by a receiver via a 'scan' operation. The representative carriers of *passive broadcast* include Bluetooth and WiFi-Direct, both of which define a mandatary 'peer discovery' scan function. Passive broadcast features negligible cost for establishing and maintaining direct links and is extremely suitable for short message dissemination in the proximity. In this paper, we present PASA with complete protocols and in-depth analysis for optimization. We have prototyped our solution on commercial phones and evaluated it with comprehensive experiments and simulation.**

I. Introduction

Smartphones have become one of the most revolutionary devices in the history of computing. With its prevalence, the scope of smartphone applications have been significantly broadened in the past few years including almost every aspect in our daily life. This paper essentially develops a local data dissemination system based on a new smartphone ad-hoc communication model. Our work is motivated by the fact that location plays an extremely important role in social networks and mobile applications. Location-based services have attracted a large volume of users [1]–[6].

The local data dissemination studied in this paper represents an attractive category of applications if communication between nearby devices is well supported. For example, a user may want to share his recent tweets or facebook messages with other people sitting in the same room; a student in library may chat with his friend in another classroom via instant messenger; a bunch of sport fans may want to share the comments with each other on the same game they enjoyed. The current location-based application architecture is still based on a centralized client-server model, where a user submits his location to a server and obtains the customized data he needs. This conventional model limits the application scope and may hinder wide deployment of location related mobile applications because of the following disadvantages. First, it requires Internet connection even when a sender and receiver are adjacently located, which will unnecessarily increase the Internet traffic burden and users' bandwidth cost. With this requirement, in addition, applications are not robust against catastrophic infrastructure failures. Furthermore, a data consumer has to gain prior knowledge of the data providers, e.g., the access to the server. There is no general channel for users to browse all available service resources nearby without registering for each and every one of them.

We argue that ad-hoc network model is a complementary alternative that can effectively help solve all the above issues. In practice, however, creating and maintaining a direct link between two nearby devices is costly. For example, both Bluetooth and WiFi-Direct require a slow initial phase of discovering nearby peers and handshaking to establish a connection. It is especially inefficient for transferring a small amount of data. In this paper, we build a local data dissemination system upon a new communication model, called *passive broadcast*. It is a connectionless and receiver-initialized model where each node periodically *scans* other nodes in the communication range and obtains their data if available (see Fig. 1, i.e., each scan is a many-to-one communication. The representative carriers of *passive broadcast* in reality include Bluetooth and WiFi-Direct, both of which define a mandatary 'peer discovery' function to fetch basic information about nearby devices. This function can be easily extended to implement passive broadcast mechanism without modifying the existing network protocol stack. In passive broadcast, the cost for establishing and maintaining direct links is negligible and our experiments show that the communication range is expanded (e.g., the communication range between two Bluetooth devices using passive broadcast is increased by 22.2m comparing to two paired devices). In addition, the feature of fetching messages from multiple nearby devices is desirable for applications that spread messages in the proximity.

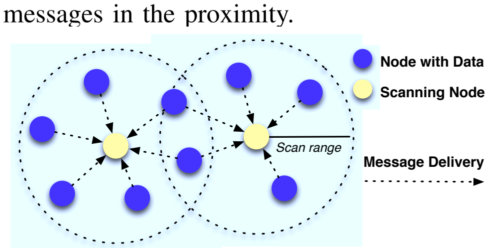


Fig. 1: Passive Broadcast Model

Our main contributions are as follows: (1) We propose to use a new communication model, passive broadcast, to carry out local data dissemination. This model is based on the method of broadcasting messages to nearby devices by overloading device names described in MDSRoB [14] and Bluejacking [23]. (2) We develop a complete local data dissemination system PASA with a suite of protocols to efficiently spread messages in the proximity. (3) We thoroughly analyze the performance and derive the best parameters to minimize the complete time. (4) We evaluate our PASA system with comprehensive experiments and simulation.

II. Related Work

Information dissemination has become important in mobile social networks. Prior work [7]–[9] aims at data dissemination in resource-constrained opportunistic networks. Messages are broadcast by superusers and ferried in intermittently connected mobile networks. Point&Connect [10] aims at simplifying the pairing process with pointing gestures of moving one device towards another. However, pairing is the initial phase for the communication. On the other hand, Musubi [11] provides a social sharing platform which enables users to share and interact with friends on the phone without having to give up privacy to any third-party service providers. From different aspects, BubbleRap [12] trying to improve the routing in Pocket Switched Networks by utilizing group membership information. And 7DS [13] tries address network disruption problem in mobile networks by providing store-carry-forward communication. In MDSRoB [14], the author proposes to use Bluetooth name to disseminate messages which is the building block for our extended passive broadcast model.

In addition, there is other prior work that helps better understand characteristic of DTNs such as [15]–[18]. However, the focus of our problem is based on a different communication model and our objective is different.

One project closely related to our passive broadcast model is called Dythr [19] which lets a phone broadcast a WiFi hotspot with the SSID being the message. This method actually is from the opposite direction of 'active' delivery as every node frequently injects messages into the wireless channel. In [20], Huang et al. propose PhoneNet which uses a central server to establish links between devices connected to a WiFi network and then allows devices in the same local network to connect directly. In [21], the authors use Bluetooth service discovery protocol to find common interests between two users. However, this work is for two-user communication while our problem is set in a multiple user environment and our goal is to determine each device's schedules to achieve the efficiency.

III. System Model

Our target problem in this paper is how to enable nearby smartphone users to share information. In this section, we introduce basic components and sketch of our solution.

A. Communication Model

Our solution is based on the new *passive broadcast* communication model. In this model, when having a message to deliver, a node puts it in a local buffer, but does not control when the message will be sent to another node. Meanwhile, each node periodically *scans* other nodes in the communication range and obtains the data in their buffers if available, i.e., each scan is a many-to-one communication.

We have implemented this model based on the mandatary 'peer discovery' function in both Bluetooth and WiFi-Direct. In the rest of this paper, we take Bluetooth as a platform instance to introduce our solution. Basically, we use the field of 'device name' to carry target payload data. The Bluetooth device name is user-specified and supposed to be exposed to remote devices. According to the standard [22] and our experiments, our solution does not affect normal Bluetooth functions. When carrying payload messages, a device can still be discovered by other devices. For a paired device, changing its device name has no impact on the communication because after pairing, the device uses logical transport address and parked member address as an identifier for data transmission.

In the PASA system, when a user intents to send a message, he assigns the message to his phone's Bluetooth device name. When other phones conduct peer discovery, the message will be sent over. The length of device names is usually limited, e.g., a Bluetooth's device name in Android can be up to 248 bytes. A large message can be fragmented to fit in and a phone can periodically change the device name to rotate multiple messages or fragments.

B. Smartphone Operations and States

In our problem setting, we assume that there are n smartphone nodes ($\{p_1, p_2, \ldots, p_n\}$) and we consider a fully-connected network model where all the phones are within each other's Bluetooth communication range. There are two basic operations for each smartphone p_i, *scan* and *update message*. The first operation is the regular peer discovery process that collects messages from other phones, and the second one is to change its own device name to a new message. *Update message* operation can be finished instantly. But *scan* process has a long overhead. For Bluetooth, according to the standard and our experiments, a scan operation usually takes $10 \sim 12$ seconds to finish. Therefore, we further define two states for each phone: when a phone is conducting *scan* (peer discovery), it is in **scanning phase**; otherwise, it is in **idle phase**. For each message, we define its *active period* as the duration when the message is available for scanning, i.e., after the message is put on the device name and before it is replaced by the next message. If a phone finishes a complete scan during a message's active period, the message will be surely scanned by the phone. If the active period starts or ends during a scanning process, the phone has a certain probability to receive the message. We will present detailed analysis later in Section V.

Specifically, we use T_i to represent p_i's *scan interval* which is defined as the interval between the end of the prior scan and the beginning of the next scan, i.e., the length of p_i's idle phase. Additionally, we use U_i to indicate the message *update interval* of p_i, i.e., p_i changes its device name once every U_i time units. Different from T_i, U_i's value can be dynamically changed as it does not incur any extra computation overhead or power consumption. Furthermore, we define S as the length of scanning phase, which is a constant for all phones.

In our solution, each phone and each active message has a unique identifier defined as follows:

- Phone ID: In this paper, we use the MAC address as each phone's identity. When scanning nearby devices, a phone automatically obtains their MAC addresses and is able to recognize these phones later.
- Message ID: Each message can be identified by its owner's MAC address and a local index number. For example,

aa:bb:cc:dd:ee:ff.12 represents a message from the phone with a MAC address of aa:bb:cc:dd:ee:ff and its index number on that phone is 12. In our solution, each index number is incremental with new messages and set up to 255 after which it will be reset to 0.

C. Message Format

In our solution, the device name is divided into two segments, header and payload, where 'header' field contains control and management data and 'payload' stores the messages being broadcast. The header includes the following fields:

- Scan interval T: Each phone p_i uses one byte to represent its own scan interval T_i in the unit of second.
- Index range of active messages: Each phone uses two bytes to specify the index number range of its active messages. We assume each user defines his own policy for disseminate the most recent messages, e.g., the most recent 10 messages, or the messages generated in the past 5 hours.
- Message reception feedback: It indicates each phone's current state of the reception. Our solution uses a hash table to represent this feedback information. The keys are the neighboring devices' phone IDs and for each neighbor phone, the associated element is a bitmap corresponding to the active messages of the neighbor. Namely, every message from a neighbor node is flagged by one bit in the feedback, '1' indicates 'received' and '0' means 'absent'. An example is shown in Fig. 2.

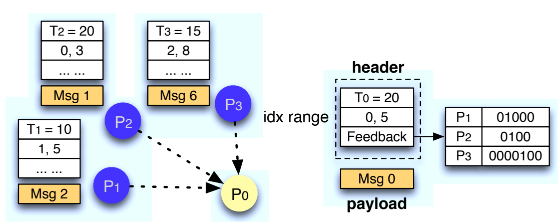


Fig. 2: p_0 just scanned its neighbors $\{p_1, p_2, p_3\}$. The header includes scan interval T_0, index range $[0, 5]$, and feedback hash table.

Table I lists some notations we use in the rest of the paper.

n/p_i	number of smartphones/the i-th phone
T_i/U_i	scan interval / update interval of p_i
S	execution time of one scan
t_i	execution time for p_i to receive all messages
f	a function that converts a MAC address to a numerical value
k_i	number of messages generated by p_i
M	set of all message IDs in the network

TABLE I: Notations

IV. Message Dissemination with Passive Broadcast

In this section, we present our solution PASA that disseminates local messages based on passive broadcast model. We further provide numerical analysis to derive the optimal parameters for our solution.

A. Problem Formulation

Recall that we consider all the phones $\{p_1, p_2, \ldots, p_n\}$ are within each other's communication range. Each phone p_i holds k_i messages to propagate to other phones. Assume each smartphone is aware of other phones' scan intervals after an initial scan. Without loss of generality, we sort all smartphones in the ascending order of their scan intervals, i.e., $\forall i, j \in [1, n]$, if $i < j$, then $T_i \leq T_j$. Let t_i represent the time phone p_i spends in receiving all the messages. Our objective in this paper is to minimize total complete time $\sum_i t_i$.

In this problem setting, both scan interval T_i and update interval U_i are important parameters for the objective. T_i, however, is also the critical parameter for energy consumption which is another important performance metric for mobile devices. Therefore, in this paper, we assume that the scan schedule (decided by T_i) has been pre-configured by each phone based on its own performance concerns, e.g., power consumption and urgency of getting new messages. The essential goal of our algorithm is to determine the update interval (U_i) for each phone so that all phones can collaborate to disseminate all messages quickly. Our solution is based on an important property defined in the following Theorem 1.

Theorem 1: If a phone set its update interval to be $T_n + 2 \cdot S$, its message can surely be scanned by all other phones.

Proof: For any phone p_i ($T_i \leq T_n$), apparently there must be a complete scan during a period of $T_n + 2 \cdot S$. Thus, the message will certainly be scanned by every phone. ■

B. Design of PASA

We present a solution with two stages for each participating phones. In the first stage, every phone simply sets U_i to be a constant α and rotates its own k_i messages. In the second stage, each phone set $U_i = T_n + 2 \cdot S$ and iteratively update the device name by one of the messages generated by its own or received from the first stage. In the first stage, the message update strategy is determined, but we need to derive the value for the update interval α. In the second stage, the update interval is fixed, and we shall design an algorithm to select the next message. In the rest of this section, we first present our algorithms for the second stage and then analyze the optimal value of α in the first stage.

In the second stage, the strategy of updating messages varies depending on each phone's status. We classify all phones into two categories and each category executes a different algorithm. The first category is called *complete phones* which includes all the smartphones that have received all $\sum_{i \in [1,n]} k_i$ messages. The other category, *incomplete phones*, is the complement of the first category and includes the phones that have not received all messages. In the rest of this subsection, we develop two algorithms, one for incomplete phones and the other for complete phones.

Our basic intuition is to let all the smartphones assign the *most wanted* messages to the device names so that they can help other phones to speed up their message collection. In addition, we intent to avoid duplicate messages in the second stage. It is apparently inefficient if multiple phones put the same message on their devices' names. In the stretch of our solution, we give incomplete phones higher priority to select messages as the messages that incomplete phones can contribute in the second stage are limited. Complete phones, on the other hand, will estimate incomplete phones' choices and select other desired messages to serve in the second stage.

We use M to represent the set of all message IDs in a given network, i.e., M includes $\sum_i k_i$ items. Assume all message IDs in M are sorted according to a pre-defined numerical conversion and let m_i represent the i-th item in M. Each phone is aware of M after the initial scan and maintains a two dimensional matrix MR to indicate the message reception status in the network, where $MR_{ij} = 1$ if phone p_i has received message m_i (otherwise $MR_{ij} = 0$). This matrix MR is built upon the feedbacks from other phones. With the assistance of MR, each phone can derive sufficient information for updating messages in the second stage. First, each phone can identify all the complete phones and incomplete phones, i.e., p_i is a complete phone if $\sum_j MR_{ij} = |M|$. Let IP and CP respectively represent the set of incomplete phones and complete phones each phone is aware of. Second, each phone knows which messages it has received are needed by other phones. We let each phone p_i build a set of *candidate messages*, indicated by C_i, including all the messages that are useful for some other phones and could be put on the device name in the next round. By definition, each message m_j must satisfy $MR_{ij} = 1$ and $\sum_h MR_{hj} < n$.

In addition, we assume each phone has a different *priority* value based on the phone ID. Our algorithm will use this priority value to avoid unnecessary duplicate message selections from multiple phones. If a message is chosen by multiple phones, only the one with the highest priority will put it on the device name and the others have to yield and select other messages. Specifically, we assume a pre-defined the function f could convert a phone ID to a numerical value for comparison, i.e., p_i has a higher priority than p_j if $f(p_i) > f(p_j)$.

Algorithm 1 Choose Message for Incomplete Phone p_i

1: Identify all incomplete phones (construct IP)
2: Sort IP in the descending order of each phone's priority
3: **for** each incomplete phone p_u **do**
4:     Construct its candidate set C_u
5:     Sort all candidate messages in C_u in the ascending order of the number of phones having received the messages, i.e., for each message m_v, the value of $\sum_j MR_{jv}$.
6: **end for**
7: **for** $u = 1$ to $|IP|$ **do**
8:     Let p_a be the u-th phone in IP, choose the first message of $C_a : C_a[1]$
9:     **for** $v = u + 1$ to $|IP|$ **do**
10:         Remove $C_a[1]$ from the v-th phone in IP
11:     **end for**
12: **end for**
13: $msg_i = C_i[1]$

Message update for incomplete phones: In our solution, an incomplete phone applies the following Algorithm 1 in the second stage. In Algorithm 1, each incomplete phone first identifies all incomplete phones (the set IP) and sorts them based on their priority values. Additionally, each phone forms the candidate sets of all incomplete phones using the information from MR (Lines 3–6). The candidate messages are also sorted by the number of phones that have received them (Line 5). In our algorithm, therefore, the first message in each candidate set is the *most wanted* messages. The main message selection process is in Lines 7–12. The loop starts from the phone with the highest priority to the one with the lowest priority. Within the loop, each phone (p_a) picks the first message ($C_a[1]$) in the candidate set for updating message in the second stage. Accordingly, all other incomplete phones with lower priorities update their candidate sets by removing $C_a[1]$ to avoid redundant message selections. Eventually, for a phone p_i, $C_i[1]$ will be put on the device name. The time complexity of Algorithm 1 is bounded by $O(n^2)$.

Message update for complete phones: The algorithm for a complete phone to update message is similar to Algorithm 1. Due to the page limit, we omit the pseudo-codes here. Each phone needs to construct the set of all complete phones CP and derive the candidate message set for each of them. An extra step for a complete phone is that it has to execute Algorithm 1 before making its own choice. All the messages that have been selected by incomplete phones will be eliminated from complete phones' candidate sets. The remaining messages in the candidate set are also sorted according to the number of phones that need them. Finally, each phone selects the first message in the sorted list of candidate messages.

C. Parameter Optimization

In this subsection, we analyze the performance of the solution presented above and derive the optimal value of α for the first stage. Essentially, we aim to express the objective as a function on α. Our analysis is based on a general function $\mathcal{P}(x, y)$ defined to represent the probability that a phone with scan interval of x can receive a message from another phone with update interval of y. In another word, $\mathcal{P}(T_i, U_j)$ is the reception probability for p_i to receive a message from p_j. We will present how to calculate $\mathcal{P}(x, y)$ in Section V. We use the following Theorem to estimate the execution time for each phone to receive all messages, i.e., the value of t_i.

Theorem 2: The execution time for phone p_i to receive all messages is expected to be $t_i = kmax \cdot \alpha + r \cdot (T_n + 2 \cdot S)$, where $kmax$ is largest value of k_i, and let $c_1 = \sum_{k_i<kmax} \frac{(kmax-k_i)\cdot\alpha}{T_n+2\cdot S}$, $c_2 = (1-\prod_i \mathcal{P}(T_i, \alpha))\cdot(1-\frac{c_1}{\sum_i k_i})\cdot \sum_i k_i$, r is the minimal value that satisfies

$$\sum_{j\in[0,r-1]} \frac{(n-1)^2}{c_2-(n-1)\cdot j} = (1-\mathcal{P}(T_i,\alpha))\cdot\sum_{j\neq i} k_j - \frac{c_1}{n}$$

Proof: Let us consider the time point when the last phone finishes its first stage, i.e., the phone with the most messages. Let $kmax = \max\{k_i\}$ Each phone p_i has missed $(1-\mathcal{P}(T_i,\alpha))\cdot\sum_{i\neq j} k_j$ messages from their owners (while in the first stage). However, some phones have broadcast messages in their second stage which guarantees a success delivery at all other nodes. In total, there have been $c_1 = \sum_{k_i<kmax} \frac{(kmax-k_i)\cdot\alpha}{T_n+2\cdot S}$ messages in the second stage. We

assume they evenly contribute to each phone's collecting process, i.e., each phone obtain at least $\frac{c_2}{n}$ missing messages.

For each message m, the probability that it has not been scanned by all the phones when the last phone enters the second stage is $(1-\prod_i \mathcal{P}(T_i,\alpha))\cdot(1-\frac{c_1}{\sum_i k_i})$, where the first term is the probability that m has not been scanned by all phones in the first stage and the second term is the probability that m is not one of the c_1 messages broadcast in the second stage. Therefore, there are $c_2 = (1-\prod_i \mathcal{P}(T_i,\alpha))\cdot(1-\frac{c_1}{\sum_i k_i})\cdot\sum_i k_i$ total candidate messages that can be put on device names in the second stage. Assume each phone picks a distinct message in the first round of second stage. For a particular message that p_i needs, there is a probability of $\frac{n-1}{c_2}$ to be scanned after the first round. In the second round, the expected number of total candidate messages becomes c_2-n and each message p_i needs has a probability of $\frac{n-1}{c_2-(n-1)}$ to be collected. This process is repeated until p_i obtains all the messages. Therefore, p_i is expected to use r rounds in the second stage to collect all messages, such that

$$\sum_{j\in[0,r-1]} \frac{n-1}{c_2-(n-1)\cdot j}\cdot(n-1) = (1-\mathcal{P}(T_i,\alpha))\cdot\sum_{j\neq i} k_j - \frac{c_1}{n}.$$

Eventually, the execution time for phone p_i is $t_i = kmax\cdot\alpha + r\cdot(T_n+2\cdot S)$. ■

Since the value of α is upper-bounded by $T_n+2\cdot S$, we can enumerate all possible values and derive the best value leading to the minimum $\sum_i t_i$ according to the above Theorem 2.

V. Analysis of Message Reception Probability

In this section, we analysis the message reception probability $\mathcal{P}(x,y)$ which is a critical building block for deriving the optimal parameters in our solution.

Based on Theorem 1, $\mathcal{P}(x,y) = 1$ if $y \geq x+2\cdot S$. The following analysis is under the condition of $y < x+2\cdot S$. The message reception probability depends on ont only the two input parameters x and y which indicate the length of scan interval and update interval (the message's active period), but also the schedule of scanning and update processes, i.e, the start points of the phone's (p_i) scanning phase and the message's (m) active period. Therefore, we need to consider several basic possible cases as follows.

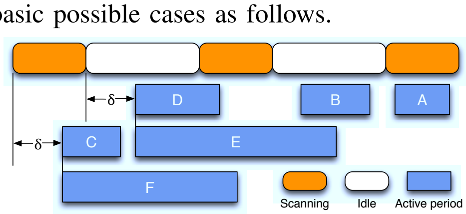


Fig. 3: Cases A∼F

Case A: The active period of m is within p_i's scanning phase. The probability that p_i can receive m is $\frac{y}{S}$.

Case B: The active period of m is within p_i's idle phase. The probability that p_i can receive m is 0.

Case C: The active period of m starts during p_i's scanning phase and ends in the consecutive idle phase. In this case, the reception probability depends on the different between the beginning of the scanning phase and that of m's active period. Let $\delta \in [0,S]$ indicate that offset. Thus, the probability that p_i can receive m is $1-\frac{\delta}{S}$.

Case D: The active period of m starts during p_i's idle phase and ends in the consecutive scanning phase. Similar to Case C, the reception probability depends on overlap of the scanning phase and m's active period. Let $\delta \in [0,S]$ be the difference between the start of the scanning phase and the end of the active period. The probability that p_i can receive m is $\frac{\delta}{S}$.

Case E: The active period of m starts in p_i's idle phase and ends in p_i's next idle phase. In this case the reception probability is 1.

Case F: The active period of m starts in p_i's scanning phase and ends in p_i's next scanning phase. In this case, p_i could possibly receive m in both scanning phases. Let $\delta \in [0,S]$ represent the different between the start of the first scanning phase and that of m's active period. Then the probability that p_i receives m in the first scanning phase is $1-\frac{\delta}{S}$. In the second scanning phase, the overlap with the active period, i.e., between the start of the second scanning phase and the end of m's active period, is $y-x-(S-\delta)$. Thus, the probability that p_i receives m in the second scanning phase is $\frac{y-x-(S-\delta)}{S} = \frac{y-x+\delta}{S}-1$. Combing these two scanning phases, therefore, the probability that p_i can receive m (in at least one of the scanning phases) is

$$\begin{aligned} & 1-(1-(1-\frac{\delta}{S}))\cdot(1-(\frac{y-x+\delta}{S}-1)) \\ = \quad & 1-\frac{\delta}{S}\cdot(2-\frac{y-x+\delta}{S}) \end{aligned}$$

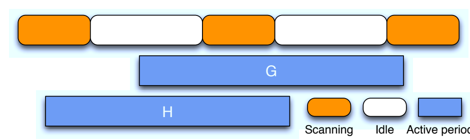


Fig. 4: Case G and Case H

Case G: The active period of m starts in p_i's idle phase and ends in the second consecutive scanning phase. The reception probability is obviously 1.

Case H: The active period of m starts in p_i's scanning phase and ends in the second consecutive idle phase. The reception probability is also 1.

Above all, we derive the value of $\mathcal{P}(x,y)$ under three different conditions in the following Theorem 3, Theorem 4, and Theorem 5. The proofs are presented in Appendix.

Theorem 3: When $y \in [x+S, x+2\cdot S)$

$$\mathcal{P}(x,y) = 1 - \frac{1}{x+S}\sum_{\delta\in[0,x+2\cdot S-y)} \frac{\delta}{S}\cdot(2-\frac{y-x+\delta}{S}))$$

Theorem 4: When $y \in [x, x+S)$

$$\begin{aligned} \mathcal{P}(x,y) = & \frac{1}{x+S}(\sum_{\delta\in[0,v)} \frac{min\{S-\delta, y\}}{S} \\ & + \sum_{\delta\in[v,S)} (1-\frac{\delta}{S}(2-\frac{\delta+y-x}{S})) + x - \frac{v^2+1}{2\cdot S}. \end{aligned}$$

Theorem 5: When $y \in (0, x)$

$$\mathcal{P}(x,y) = \frac{1}{x+S}\Big(\sum_{\delta\in[0,v)} \frac{min\{S-\delta, y\}}{S} + \sum_{\delta\in[v,S)} (1 - \frac{\delta}{S}(2 - \frac{\delta+y-x}{S}))\Big).$$

VI. Implementation and Evaluation

A. System Implementation

We choose Android operating system as our development environment, and have implemented PASA on different brands of smartphones and tablets, including Nexus 7, Nexus 4 and Samsung Galaxy Nexus. Our prototype system is built on Android Jelly Bean(specific version 4.2.2), API level 17. PASA system does not require any change on Bluetooth driver or Android kernel.

In our experiments, we integrate PASA with Twitter to demonstrate application functions. A user can choose to pull online tweets for local dissemination or push local messages to his Twitter account. As a result, when a user starts PASA for the first time, he needs to authorize the application to use his Twitter account and configure his own profile (e.g., name, number of messages to share and length of idle duration). Fig. 5 and Fig. 6 show the user interfaces of our PASA system.

The system mainly performs the following operations: (1) input and publish message to Twitter, nearby users or both (Twitter, BLocal and BTwitter on Fig. 5 correspondingly), (2) set or change configurations, (3) scan the nearby users to receive messages, (4) generate and transmit feedback to inform other users which messages it currently holds, and (5) view one specific local user's shared message.

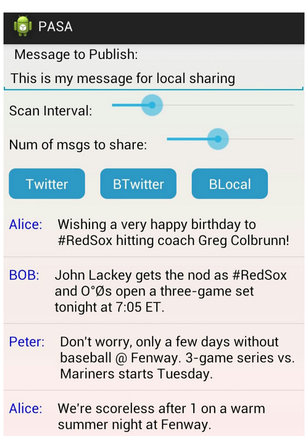


Fig. 5: PASA: Main page

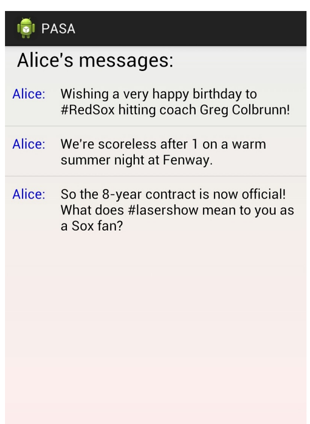


Fig. 6: PASA: Message list

In addition, we develop a simulator to evaluate a system with a large number of smartphones. For both experiments and simulation, a scanning phase lasts 10 seconds ($S = 10$).

B. Performance Evaluation

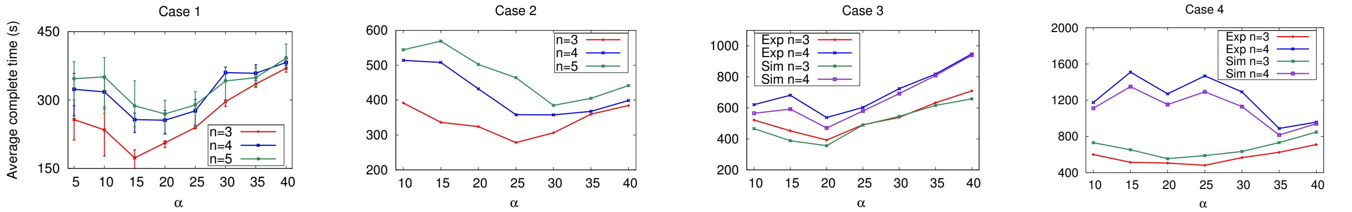
In this subsection, we present evaluation results from both Android smartphones experiments (small scale) and simulation (large scale). The major performance metric we exam is the average complete time among all smartphones in the network (which is equivalent to the objective function in our algorithm design $\sum_i t_i$).

1) Experimental Results: We set up a small testbed with 5 Android phones running our program. We consider the following four different cases for our evaluation. **Case 1**: All smartphones have the same scan interval(T_i) and number of messages(k_i); **Case 2**: They have different T_i but the same k_i; **Case 3**: They have different k_i but the same T_i; **Case 4**: Both T_i and k_i are different.

Fig. 7a shows the average complete time in Case 1 with varying α. In this setting, each smartphone has 10 messages($k_i = 10$) to share and their scan intervals is 5s($T_i = 5$). We change α from 5s to 40s with a interval of 5s. For each parameter setting, we repeat the experiment for 5 times and present the average value in Fig. 7a. The error bar indicates the maximum and minimum complete time observed in our tests. As shown in the figure, with 3 phones, the best value of α is 15; with 4 or 5 phones the best results are achieved when $\alpha = 20$. Intuitively, a smaller α causes low message reception rate in the first stage leading to more messages disseminated in the second stage; on the other hand, a larger α increase the execution time of the first stage. Therefore, there are pivot points in all three curves. Our analytical model derives the optimal values of α to be 18, 19, 19 respectively for $n = 3, 4$ and 5. This estimation matches well with the experimental results which shows our analysis is accurate and effective, thus can help users find the best value of α for the PASA system. Fig. 7b shows a result for Case 2, where $k_i = 10$ and T_i is set to 10, 15, 20, 25, 30 respectively. Based on the experiments, the best tested values of α are 25, 30, 30 for $n = 3, 4, 5$. Our theoretical model derives 26, 26, and 30 as the optimal values which are very close to the experimental results.

Fig. 7c and Fig. 7d illustrate the experimental results for Case 3 and Case 4 in our evaluation. In addition, we conduct simulation with the same setting to confirm the accuracy of our own simulator. We use 'Exp' and 'Sim' in the legend to represent the experimental and simulation results respectively. In Fig. 7c, $T_i = 10$ for all phones and k_i is set to 10, 15, 20, 25 for each phone respectively. In the tested cases, the best value of α with minimum average complete time is 20 for both $n = 3$ and $n = 4$. Our analysis again derives very close values of 23 and 24 for $n = 3$ and $n = 4$. Additionally, our simulation results matches the experimental result very well with small gaps. In Fig. 7d, we show the results of different T_i and k_i. Given a particular value of α, the average complete time in the experiment with $n = 4$ is much larger than $n = 3$. Comparing with $n = 3$, the network with $n = 4$ has $T_4 = 25$ which results in large difference of average complete time. Phone p_4 with $T_4 = 25$ is more likely to miss messages during the first stage and thus needs more time to collect messages during other phones' second stage. Our theoretical optimal values of α are 25 and 35 for $n = 3$ and $n = 4$ which match the results in Fig. 7d very well.

2) Simulation Results: To evaluate a large scale network, we conduct simulation to test the performance of PASA. Our



(a) $T_i = 5$ and $k_i = 10$

(b) $k = 10$ for all smartphones and $T_i = 10, 15, 20, 25, 30$

(c) $T_i = 10$ for all smartphones and $k_i = 10, 15, 20, 25$

(d) $T_i = 10, 15, 20, 25$, $k_i = 10, 15, 20, 25$

Fig. 7: Average complete time with different α

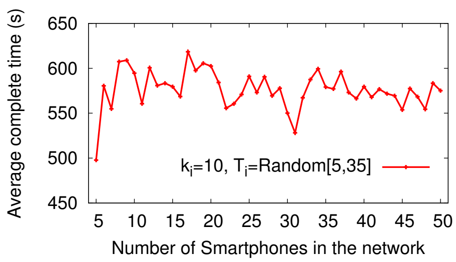
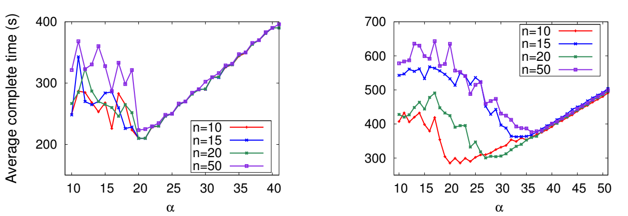


Fig. 8: Average complete time with different n, while $k_i = 10$ for all smartphones and $T_i = Random[5, 35]$

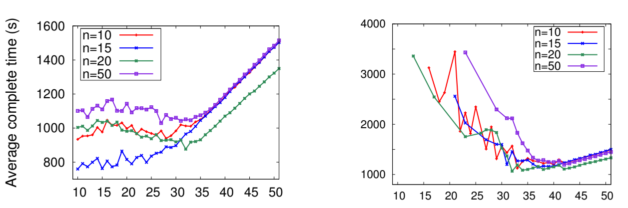
goal is to study the impact of the number of phones on the system performance. Fig. 8 shows the average complete time with $k_i = 10$ and $T_i = Random[5, 35]$ over different number of smartphones in the network. The number of phones ranges from 5 to 50, and the average complete time fluctuates between 500 and 600 which is very small comparing with the number of total messages increase. This is caused by the many-to-one communication during each scan. Increasing the number of phones does not increase the average complete time much which indicates passive broadcast is certainly a scalable model.



(a) $T_i = 10, k_i = 10$ for all smartphones

(b) $k_i = 10$ for all phones and $T_i = Random[5, 35]$

Fig. 9: Average complete time with different α



(a) $k_i = Random[5, 30]$ and $T_i = Random[5, 30]$

(b) With message rotation (Comparison to Fig. 10a)

Fig. 10: Average complete time with different α

Fig. 9a shows the average complete time for 10, 15, 20, and 50 smartphones. We find that for $\alpha \geq 30$, the five curves are almost coincide with each other. The reason is that α already reaches $T_n + 2 \cdot S$. According to Theorem 1, this update interval is sufficiently long such that every other phone can certainly receive every message during the first stage. From our theoretical model, the optimal value for these settings are 24, 26, 22, 23 for $n = 10, 15, 20, 50$. Comparing to Fig. 9a, our model can yield a good value of α for PASA system in these settings. Fig. 9b shows the result when $k = 10$ for everyone and each phone randomly chooses T_i in $[5, 35]$. In our tested cases, for $n = 10, 15, 20, 50$ the maximum T_i is 29, 28, 21, 35 respectively. The theoretical optimal values of those four settings are 20, 26, 25, 33. Finally, Fig. 10a shows a general scenario where a user's k_i and T_i are randomly chosen from [5, 30]. The following table shows the specific setting of our test. With the details of each setting, our theoretical model gives the optimal value of α as 29, 25, 26, 34.

	$n = 10$	$n = 15$	$n = 20$	$n = 50$
$\max(k_i)$	30	30	27	30
$\max(T_i)$	29	30	28	30

Besides showing the accurate estimation of α, we further present a comparison to a simple message rotation scheme to show the advantage of our two-stage algorithm. In the rotation scheme, each phone rotates its own messages cyclically with the update interval α, i.e., repeating the first stage defined in our solution. We conduct a test with exactly the same parameter setting with Fig. 10a and the results are illustrated in Fig. 10b. Note that we set a upper bound of 10,000 seconds for the execution time in our simulation and the missing values in Fig. 10b indicate that the simulation for those settings exceeds the upper bound of the execution time. Comparing to Fig. 10a, we observe that our two-stage PASA system significantly reduces the average complete time especially when α is small. Our solution is more robust to a misconfigured α. Even when α is not set to the optimal value, PASA system can mitigate the negative impact on the performance.

VII. Conclusion

This paper develops PASA, a local message dissemination system based on *passive broadcast* model. We present a two-stage protocol for propagating messages and provide in-depth analysis to derive the optimal update interval value. We have implemented our solution on Android phones and evaluated it with experiments and simulation. The results show that PASA is effective and efficient for disseminating local messages.

References

[1] Foursquare. http://www.foursquare.com.

[2] Facebook Places. http://www.facebook.com.
[3] Yelp. http://www.yelp.com.
[4] Waze. http://www.waze.com.
[5] SCVNGR. http://www.scvngr.com.
[6] UBER. http://www.uber.com.
[7] Chiara Boldrini, Marco Conti, and Andrea Passarella. Contentplace: social-aware data dissemination in opportunistic networks. In MSWiM '08, 2008.
[8] Jialu Fan, Jiming Chen, Yuan Du, Wei Gao, Jie Wu, and Youxian Sun. Geocommunity-based broadcasting for data dissemination in mobile social networks. *IEEE Trans. Parallel Distrib. Syst.*, 24(4), April 2013.
[9] Bahadir K. Polat, Pushkar Sachdeva, Mostafa H. Ammar, and Ellen W. Zegura. Message ferries as generalized dominating sets in intermittently connected mobile networks. In MobiOpp '10, 2010.
[10] Chunyi Peng, Guobin Shen, Yongguang Zhang, and Songwu Lu. Point&Connect: intention-based device pairing for mobile phone users. In MobiSys '09, 2009.
[11] Ben Dodson, Ian Vo, T.J. Purtell, Aemon Cannon, and Monica Lam. Musubi: disintermediated interactive social feeds for mobile devices. In WWW '12, 2012.
[12] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11), November 2011.
[13] Arezu Moghadam, Suman Srinivasan, and Henning Schulzrinne. 7ds - a modular platform to develop mobile disruption-tolerant applications. In *Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services and Technologies*, 2008.
[14] Joseph Paul Cohen. Wireless message dissemination via selective relay over bluetooth (mdsrob). *CoRR*, abs/1307.7814, 2013.
[15] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Multicasting in delay tolerant networks: a social network perspective. In MobiHoc '09.
[16] Mooi Choo Chuah. Social network aided multicast delivery scheme for human contact-based networks. In SIMPLEX '09, 2009.
[17] Josep Díaz, Alberto Marchetti-Spaccamela, Dieter Mitsche, Paolo Santi, and Julinda Stefa. Social-aware forwarding improves routing performance in pocket switched networks. In *Proceedings of the 19th European conference on Algorithms*, ESA'11, pages 723–735, Berlin, Heidelberg, 2011. Springer-Verlag.
[18] Elizabeth M. Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In MobiHoc '07, 2007.
[19] S. Jakubczak. Dythr. http://szym.net/dythr/.
[20] Te-Yuan Huang, Kok-Kiong Yap, Ben Dodson, Monica S. Lam, and Nick McKeown. PhoneNet: a phone-to-phone network for group communication within an administrative domain. In MobiHeld '10.
[21] Adam C. Champion, Zhimin Yang, Boying Zhang, Jiangpeng Dai, Dong Xuan, and Du Li. E-smalltalker: A distributed mobile system for social networking in physical proximity. *IEEE Trans. Parallel Distrib. Syst.*, 24(8), August 2013.
[22] Bluetooth Link Manager Protocol (LMP) Architecture. http://developer.bluetooth.org/TechnologyOverview/Pages/LMP.aspx.
[23] Bluejacking. http://en.wikipedia.org/wiki/Bluejacking

APPENDIX A
PROOF OF THEOREM 3

Proof: We consider a scanning phase followed by a idle phase as a repeating round for phone p_i and the message active period could start at any point in this round. When $y \in [x+S, x+2\cdot S)$, as shown in Fig. 11, we can classify all possible scenarios into two conditions based on the starting point of m's active period. In Condition 1, the active period starts in p_i's scanning phase with a offset from the beginning of the scanning phase to a point where the active period ends at the end of the next scanning phase.

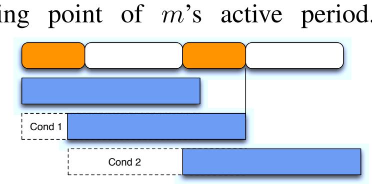


Fig. 11: $y \in [x+S, x+2\cdot S)$

Condition 2 is the complementary of Condition 1. Apparently, the reception probability under Condition 2 is 1 because the active period covers a complete scanning phase (Case H). On the other hand, Condition 1 refers to Case F above and the difference from the start of the scanning phase ranges from 0 to $u = x+2\cdot S-y$. Therefore, the reception probability $\mathcal{P}(x,y)$ is

$$\begin{aligned} & (1-\tfrac{u}{x+S}) + \tfrac{1}{x+S}\textstyle\sum_{\delta\in[0,u)}(1-\tfrac{\delta}{S}\cdot(2-\tfrac{y-x+\delta}{S}))) \\ = \ & 1-\tfrac{1}{x+S}\textstyle\sum_{\delta\in[0,u)}\tfrac{\delta}{S}\cdot(2-\tfrac{y-x+\delta}{S})) \end{aligned}$$

■

APPENDIX B
PROOF OF THEOREM 4

Proof: Let SP_1 and SP_2 be the closest scanning phase before and after the message active period starts, i.e., SP_1 belongs to the round, but SP_2 does not. Under the condition of $y < x+S$, the message active period could overlap with either SP_1 or SP_2 or both of them. Let δ be the offset of the message active period compared to the start of a round. We derive $\mathcal{P}(x,y)$ based on the analysis of the following three cases: (1) The message active period only overlaps with SP_1, but not SP_2. This case refers to Case A and Case C and δ ranges from 0 to $S+x-y$, the message reception probability is $\frac{min\{S-\delta,y\}}{S}$. (2) The message active period only overlaps with SP_2, but not SP_1. This case refers to Case D, Case E and Case G. The message reception probability is $\frac{\delta+y-S-x}{S}$ when $\delta \in [S, x+2\cdot S-y]$ (Case D). The message reception probability is 1 when $\delta \in [x+2\cdot S-y, x+S]$. Therefore, let $v = x+S-y$, the message reception probability is

$$\begin{aligned} & \sum_{\delta\in[S,v+S)}\frac{\delta-v}{S} + \sum_{\delta\in[v+S,x+S)} 1 \\ = \ & \frac{(v+2\cdot S-1)\cdot v}{2\cdot S} - \frac{v^2}{S} + (x-v) = x - \frac{v^2+1}{2\cdot S} \end{aligned}$$

(3) The message active period overlaps with both SP_1 and SP_2. In this case, δ ranges from $x+S-y$ to S. The lengths of the overlaps with SP_1 and SP_2 are $S-\delta$ and $\delta+y-x-S$ respectively. Referring to Case F, the message reception probability is

$$\begin{aligned} & \sum_{\delta}(1-(1-\frac{S-\delta}{S})(1-\frac{\delta+y-x-S}{S})) \\ = \ & \sum_{\delta}(1-\frac{\delta}{S}(2-\frac{\delta+y-x}{S})) \end{aligned}$$

■

APPENDIX C
PROOF OF THEOREM 5

Proof: The proof is similar to the above proof of Theorem 4. Under the condition of $y < x$, the message active period could overlap with either SP_1 or SP_2. Let δ be the offset of the message active period compared to the start of a round. We derive $\mathcal{P}(x,y)$ based on the analysis of the following two cases: (1) The message active period only overlaps with SP_1. This case refers to Case A and Case C and δ ranges from 0 to $S+x-y$, the message reception probability is $\frac{min\{S-\delta,y\}}{S}$. (2) The message active period only overlaps with SP_2. This case refers to Case D and Case E. The message reception probability is $\frac{\delta+y-S-x}{S}$ when $\delta \in [S, x+2\cdot S-y]$.

■