

Towards Connectionless Communication for Local Message Dissemination

Ying Mao, Member, IEEE

Abstract—With the prevalence of smartphones, the scope of mobile applications has been significantly broadened in the past few years including almost every aspect of our daily life. Among the applications, message dissemination services play a significant role. Traditionally, users send messages to the servers which forward them according to friend lists. This client-server architecture is efficient with the support of infrastructures, however, it restricts the application scope and may hinder wide deployment of location-based mobile applications. This paper proposes a novel communication model called *passive broadcast* for local message dissemination. 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 mandatory ‘peer discovery’ function. Passive broadcast features negligible cost for establishing and maintaining direct links and is extremely suitable for short message dissemination in the proximity. Furthermore, we study the operation management in the proposed communication model with complete protocols and in-depth analysis for optimization. We prototype our solution on commercial phones and evaluated it with extensive experiments and simulation.

Index Terms—Message Dissemination, Location-Based Services, Ad-hoc Networks, Connectionless Communication.

I. INTRODUCTION

Smartphones have become one of the most revolutionary devices in the history of computing. With its prevalence, today’s smartphones have gone far beyond a mobile telephone as they have seamlessly dissolved in every aspect of our daily life. Together with the fast-development of cloud computing [1]–[3] and edge computing [4]–[6], a variety of applications has been developed to meet all kinds of demands. For instance, Facebook [7] and Twitter [8] are popular in social networks; Google Maps [9] and Waze [10] are used for navigations; Siri [11] and Cortana [12] provide virtual assistants. Furthermore, the deployment of deep learning applications on the cloud [3], [13]–[15] has employed applications with more intelligence. These applications significantly improve the quality of our life, regarding connectivity, productivity, and efficiency. Most of the applications are built on the fundamental of a communication system, the network infrastructures, which include many components, such as cellular towers, cables, servers, switches, and routers, etc. This system along with the client-server architecture have been proved to provide stable services to hundreds of thousands of users every second. However, this conventional client-server model limits the application scope and may hinder wide deployment of location related mobile

applications because of the following disadvantages. First, there is no general channel for users to browse all available service resources nearby without registering for each and every one of them on the servers. Furthermore, 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.

Motivated by the facts that people carriers a smartphone most of the time [16], this paper develops a local data dissemination system that is based on a smartphone ad-hoc communication model. The smartphone ad-hoc network aims to provide a robust and infrastructure-free alternative to the cellular network. In practice, however, creating and maintaining a direct link between two nearby devices, which is the building block of an ad-hoc network, is costly. For example, both Bluetooth and WiFi-Direct require a slow initial phase of discovering nearby peers and handshaking to establish a connection that introduce user operations. It is especially inefficient for transferring a small amount of data locally. In addition, connected link may bring with it a security risk because one device might be able to access all exposed services on the connected device (e.g., in Bluetooth). Pairing Bluetooth devices also brings with it a security risk because because a paired device can access all exposed services on a device. In this project, we build a local data dissemination system upon a new communication model, called *passive broadcast* [17]. 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 mandatory ‘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. We particularly use the field of ‘device name’ to carry payload data. When a user intends to send a message, he assigns the message to his phone’s Bluetooth or WiFi-Direct device name. When other phones conduct the peer discovery, the message will be sent over.

Compared to traditional Mobile Ad-hoc Networks, in passive broadcast model, the cost for establishing and maintaining direct links is negligible and our experiments show that the communication range is expanded compare to the established connection over the same radio interface. How-

Ying Mao is with the Department of Computer and Information Science, Fordham University, New York, NY, 10458 Email: maoy@tcnj.edu

ever, this receiver-initialized communication inevitably causes a delay on the sender side after data is ready and before data is sent. Thus, network throughput of passive delivery is limited. In this project, we use passive delivery to build local data dissemination systems for gossip-style communication. Our target applications include spreading tweets, facebook messages or Google Hangout messages to nearby nodes and local instant messenger.

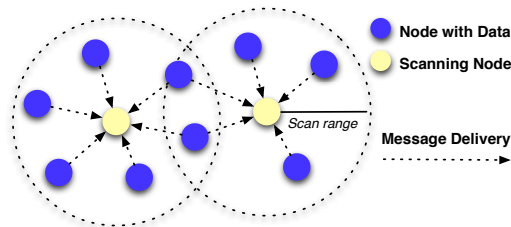


Fig. 1. Passive Broadcast Model

Our main contributions are as follows:

- We propose to use a new communication model, passive broadcast, to carry out short messages for local data dissemination.
- We develop a complete communication system with a suite of protocols and power-saving mode to efficiently spread messages in the proximity.
- We study the 'scan' management which is crucial to the proposed network model and analyze the performance to derive the best parameters to minimize the complete time.
- We prototype the system on commercial phones and evaluate it with comprehensive experiments on the testbed and simulation.

II. RELATED WORK

Ad-hoc networks have been proposed as a complementary alternative to the traditional infrastructure-based communication technology.

In the literature, several mobile ad-hoc networks (MANETs) based communication system have been proposed for various scenarios. In mobile social networks, authors in [18]–[21] aim at data dissemination in resource-constrained opportunistic networks, broadcasting from superusers and ferrying messages in intermittently connected mobile networks. Point&Connect [22] implements pointing gestures of moving one device towards another in order to enable spontaneous device pairing. Musubi [23] provides a decentralized trusted social services on personal mobile devices. While BubbleRap [24] utilizes centralized membership information to improve standard unicast routing. WiFi-Opp [25] is developed to support opportunistic communications by exploiting the mobile AP mode. However, AP mode is unsupported on low-end smartphones.

For delay tolerant networks, the authors in [26] propose a gossiping-based approach, where each node forwards a message with some probability, to reduce the overhead of the routing protocols. The protocol uses notably fewer messages than flooding and provides well enough performance. As people are moving or on the vehicle, unlike traditional network,

mobile social network is not always connected. Authors in [27] propose to use data diffusion to reduce the query delay in DTNs. They use theoretical models to analyze the data diffusion process and compare the performance of their proposed diffusion schemes in terms of diffusion speed and query delay.

In the field of opportunistic networks, through adopting the concept of opportunistic beacons, the idea of Opportunistic Beacon Networking (OBN) is presented in [28]. It is a highly-opportunistic but a limited-throughput approach. Its enhanced version, BLESSED [29], utilizes the a dual-radio setting of both WiFi Hotspot and Bluetooth low energy (BLE).

Due to the infrastructure-free feature, the ad hoc network is also widely used in disaster rescue. 7DS [30] is developed to address network disruption problem in mobile networks by providing store-carry-forward communication. Wu et al. [31] present a routing protocol to disseminate emergency messages among any smartphones when cellular infrastructure is unavailable. However, it only focuses on prioritization of messages for one-way unicast communications. DistressNet [32] allows users to quickly and reliably store and retrieve data in a cloud-like manner. Emergency response personnel can use portable devices such as tablets or smartphones to monitor field data and collaborate with other teams. However, DistressNet requires deploying large amount of sensors over the disaster area which is not available for most of the scenarios. DISANET [33] is used to exchange text, voice and video messages between rescue workers and the command center. TeamPhone [34] integrates cellular networking, ad-hoc networking and opportunistic networking seamlessly, and enables communications among rescue workers. However, on principal, it is based on Greedy Perimeter Stateless Routing (GPSR).

Although our paper is based on MANETs, the focus of the problem is a different communication model and our objective is different from the previous works. Passive broadcast is a receiver-initialized connectionless communication based on many-to-one delivery in each scan. In addition, we target on a distributed and collaborative solution that efficiently disseminate messages between rescuers nearby victims. Furthermore, the proposed system requires limited operations from the users. One project closely related to our passive broadcast model is called Dythr [35] 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 [36], Huang et al. proposed PhoneNet [37]. This method uses a central server to establish links between devices connected to a WiFi network and then allows devices connected on local networks to connect directly. In [38], the authors use Bluetooth service discovery protocol to find common interests between two users. Similar to our proposed work, no connection is established and each user stores the keyword about his interest in Bluetooth service ID. 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

The main research target of this project is to develop a local message dissemination system based on passive broadcast with the objective of achieving efficiency in terms of time overhead and power consumption. In another word, how to quickly disseminate messages in a smartphone ad-hoc network with as little power consumption as possible.

A. Communication Model

Our solution is based on the new *passive broadcast* (PASA) communication model, where each sender buffers its broadcast data locally and the receiver initializes the transmission and fetches all available data from nearby nodes. In the system, PASA serves as a middleware between applications (e.g. Facebook, Twitter) and network interfaces. When having a message to deliver, the application sends to PASA middleware. According to user's specifications, PASA decides which network interfaces to use. If the users would like to share the message online, WiFi or cellular network interface will be chosen. Otherwise, if the users would like to share the message with nearby strangers, the ad-hoc networks will be used and Bluetooth or WiFi-Direct interface will be chosen. In the later settings, the message stores in local buffer, and the application does not control when the message will be sent to another nearby 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. Fig 2 illustrates the major components in the architecture of PASA.

- **Constraint collector:** It collects the specified constraints include how urgent a user need to spread this messages or collecting nearby messages, limitations of power consumption, and a particular performance objective.
- **Workload manager:** It manages the local buffer that stores messages both generated by the users or received from others. According to the user specified constraints, each message may have different priority, e.g. user may want quickly spread its own message and delay others.
- **Feedback analyst:** It maintains a database that contains all the messages and their priorities based on the feedback from nearby users. The priority of each message needs to update along with scanning.
- **Message update scheduler:** It controls a parameter, update interval, that determines when to update the messages. Due to the small space, PASA can only spread limited number of messages at one time.
- **Message selector:** It selects the messages in the buffer based on their priorities and how large of the demand from nearby users. The message with highest priority and largest demand will be selected.
- **Scan scheduler:** Based on user's specifications, it decides when to initialize the communication and when to stay idle. A more frequently scan schedule results in a faster spreading / receiving messages and a higher cost in battery life.
- **Distributed controller:** It is a distributed and collaborative algorithm for each phone to decide how to scan nearby

peers and update messages, where the process of updating messages includes not only the schedule but also the message selection strategy.

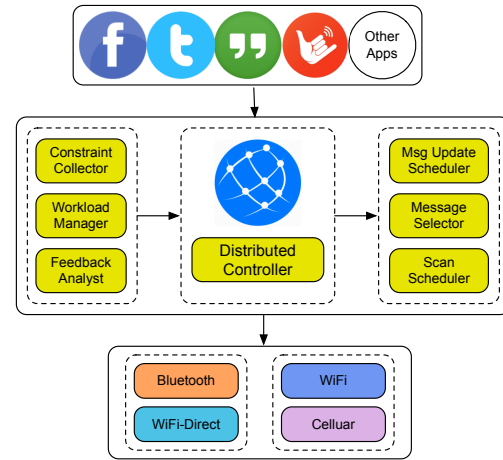


Fig. 2. System Design Architecture

We have implemented this model based on the mandatory '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. When a user intends 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 ($\{p_1, p_2, \dots, p_n\}$) 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 ~ 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. Similar to other network protocol, '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:** Each phone includes a *message reception feedback* to indicate the 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 the associated elements are a structure of bitmaps. 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. 3.

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

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.

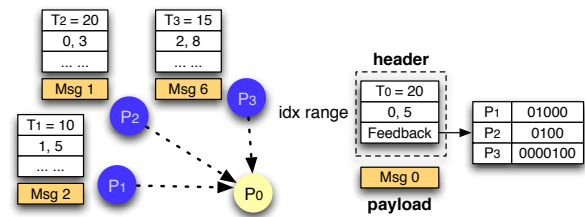


Fig. 3. Message Format: Phone p_0 has just scanned its neighbors $\{p_1, p_2, p_3\}$. The header includes the scan interval T_0 , index range $[0, 5]$, and the feedback hash table.

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 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

A. Problem Formulation

Recall that we consider all the phones $\{p_1, p_2, \dots, p_n\}$ 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., in our setting, 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. In addition, each phone 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 $MR_{ij} = 0$). For each phone, 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 $m_j \in C_i$ 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 $f(p_i) > f(p_j)$.

Message update for incomplete phones: In our solution, an incomplete phone applies the following Algorithm 1 in the second stage.

Algorithm 1 Choose the Message for Incomplete Phone p_i

- 1: Identify all incomplete phones (construct IP)
 - 2: Sort the set 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: **for** $u = 1$ to $|IP|$ **do**
 - 7: Let p_a be the u -th phone in IP , choose the first message of $C_a : C_a[1]$
 - 8: **for** $v = u + 1$ to $|IP|$ **do**
 - 9: Remove $C_a[1]$ from the v -th phone in IP
 - 10: $msg_i = C_i[1]$
-

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 , $msg_i = C_i[1]$ will be the message to 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. As same as in Algorithm 1, each phone selects the first message in the sorted list of candidate messages.

C. Network Management with Parameter Optimization

From the analysis above, the update interval(α) is crucial to the performance of PASA system. In this subsection, we present the network management through analyzing the performance of the solution 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 $c_1 = \sum_{k_i < kmax} \frac{(kmax - k_i) \cdot \alpha}{T_n + 2 \cdot S}$, $c_2 = (1 - \prod_i \mathcal{P}(T_i, \alpha)) \cdot (1 - \frac{c_1}{\sum_i k_i})$. r is the minimal value that satisfies the following condition,

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

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 $\frac{c_1}{n}$ 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

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

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.

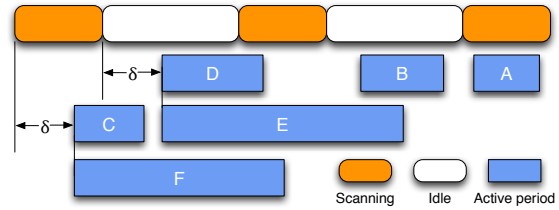


Fig. 4. Cases A~F

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. This case is similar to Case C and 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

$$1 - (1 - (1 - \frac{\delta}{S})) \cdot (1 - (\frac{y-x+\delta}{S} - 1)) \\ = 1 - \frac{\delta}{S} \cdot (2 - \frac{y-x+\delta}{S})$$

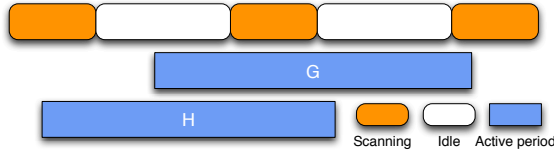


Fig. 5. Case G and Case H

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 with The details of the proof.

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

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

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.

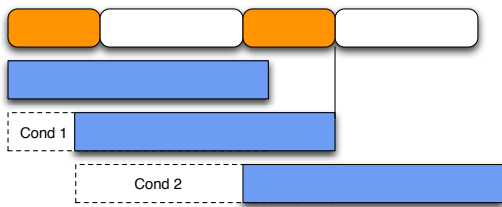


Fig. 6. $y \in [x + S, x + 2 \cdot S)$

When $y \in [x + S, x + 2 \cdot S)$, as shown in Fig. 6, 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. 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 is

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

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

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

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 $\frac{\delta + y - x - S}{S}$ 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

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

(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 Case F, the message reception probability is

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

Therefore, when $y \in [x, x + S)$,

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

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

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

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 $\delta \in [S, x + 2 \cdot S - y]$ (Case D).

Therefore, following the above proof of Theorem 4, when $y < x$,

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

VI. PROBLEM SETTING WITH OFF STATE

In the previous sections, we consider the phones' Bluetooth modules are always "ON". However, in reality, some users may want to save energy on each application to maintain a longer battery life, especially disaster victims. Thus, for each phone, it may have a energy budget. In order to meet the energy budget per unit constraint, we assume that phones can turn off their Bluetooth modules periodically. Now, we define the following three operations for each phone.

- **OFF state:** In OFF state, the Bluetooth module on the mobile device is turned off. Therefore, it cannot interact with the nearby devices to receive or deliver messages. Consequently, the system does not require any energy from the smartphone.
- **IDLE state:** In IDLE state, the smartphone can only deliver messages to others and does not receive any messages from the nearby mobile devices.
- **SCAN state:** In SCAN state, the nodes can not only initial communication to the nearby smartphones to receive messages but also response to others' scan to deliver its own messages.

Apparently, the power consumption of the Scan, Idle and OFF states are different. Clearly, during the OFF state, there is no energy cost required from the smartphone. Fig. 7 shows a sample schedule including these three operations. In each "ON" period, it can contain one or multiple Scans and Idles.



Fig. 7. Three States

A. Message Format with OFF state

In our solution, the device name is divided into two segments, header and payload. Similar to other network protocol, 'header' field contains control and management data and 'payload' stores the messages being broadcast. For a phone i , the header should includes its schedule of the state transitions and current message reception state. Besides message reception feedback and index range of active messages fields as described in section III, the message format with OFF state introduces the following new fields:

- A_i : each phone uses two bytes to indicate the time stamp when the Bluetooth device was most recently turned on and became active.
- L_i^{on} and L_i^{off} : each phone uses two bytes to illustrate the length of the Bluetooth device's active period (the device is on) and the length of its inactive period (the device is off).
- I_i : each phone uses one byte to express the time interval between two consecutive scans when the Bluetooth device is on, i.e., the gap between the end of the previous scan and the beginning of the next scan.

B. Parameter Optimization with OFF state

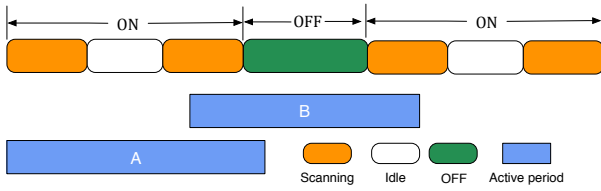
During the initial scan, phones exchange their scheduling parameters, including lengths of "ON" and "OFF" state. We assume that, for an incomplete phone, the beginning of each "ON" period is a Scan operation and the. The Scan operation has a fix length naturally and as in the previous problem setting, the Idle operation is configured to be one particular value. Therefore, after exchanging the length of each "ON", "OFF" and "Idle", phones can calculate each other's scheduling.

$$L_{scan} * N_{scan} + L_{idle} * N_{idle} = L_{ON} \quad (1)$$

where, L_{scan} is a naturally fix value for all phones, L_{idle} and L_{ON} are the length of each phone's Idle and ON states. N_{scan} and N_{idle} are the number of Scans and Idles during ON state. Since each ON state starts with a Scan operation and every Idle has a previous Scan operation, we have $N_{scan} = N_{idle}$ or $N_{scan} = N_{idle} + 1$. Therefore, we can derive the values of N_{scan} and N_{idle} .

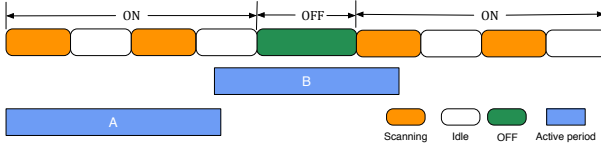
After deriving the number of Scans and Idles, we start analyzing the minimum update interval which can ensure others receive this message. As shown above, the relationship between N_{scan} and N_{idle} can be either $N_{scan} = N_{idle} + 1$ or $N_{scan} = N_{idle}$. According to the setting of ON period, $N_{scan} = N_{idle} + 1$ indicates that the ON period starts and ends with a Scan operation. If $N_{scan} = N_{idle} + 1$, it starts with a Scan and ends with an Idle operation. Now, we analyze the minimum update interval for both cases.

- $N_{scan} = N_{idle} + 1$: The message that activates for at least minimum update interval ensures the reception by

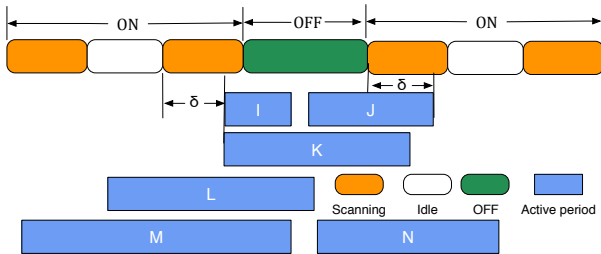
Fig. 8. Minimum update interval for $N_{scan} = N_{idle} + 1$

others. This is achieved by covering at least one complete *Scan* state. Fig. 8 shows a sample setting in this case. The analysis in the previous section shows that if the phones always stay on, $L_{idle} + 2 * L_{scan}$ is the minimum update interval as shown in Fig. 8 case A. However, it is different after we introduced “OFF” state. If the $L_{idle} + 2 * L_{scan}$, is less than $L_{OFF} + 2 * L_{scan}$, this active length can not ensure to cover at least one complete *Scan* operation, as shown in Fig. 8 case B. Therefore, in this case, the minimum update interval should be $\max\{L_{idle} + 2 * L_{scan}, L_{OFF} + 2 * L_{scan}\}$.

- $N_{scan} = N_{idle}$: In this setting, the “ON” state starts with a *Scan* and ends with a *Idle* operation. Case A, in this scenario, is the same with the previous one, however case B is different. Since “OFF” state is always behind a *Idle* operation, the case B should be $L_{idle} + 2 * L_{scan} < L_{idle} + L_{OFF} + L_{scan}$. Therefore, similar to the previous result, in this scenario, the minimum update interval should be $\max\{L_{idle} + 2 * L_{scan}, L_{idle} + L_{OFF} + L_{scan}\}$.

Fig. 9. Minimum update interval for $N_{scan} = N_{idle}$

From the above analysis, we can determine the minimum update interval after the initial scan. Besides the cases A~H that present in Fig 4 and 5, we have a 12 more cases with the new setting as shown in Fig 10 and 11.

Fig. 10. Minimum update interval for $N_{scan} = N_{idle}$

Case I: The active period of m starts during p_i 's last scanning phase and ends in the following “OFF” state. In this case, the reception probability depends on the difference between the beginning of the scanning phase and that of m 's active period. Let $\delta \in [0, S]$ indicates the offset, thus, the probability that p_i can receive m is $1 - \frac{\delta}{S}$.

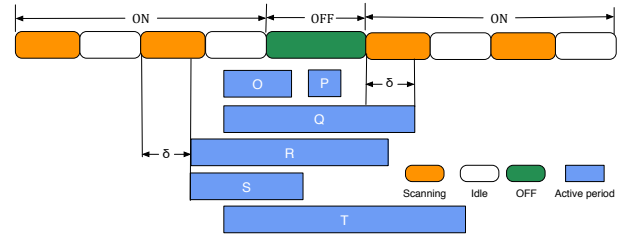
Case J: The active period of m starts during p_i 's “OFF” State and ends in the following *Scan* phase. Similar to Case I, the

reception probability depends on the overlap of *Scan* phase and m 's active period. Let $\delta \in [0, S]$ be the overlap, the reception probability that p_i can receive message m is $\frac{\delta}{S}$.

Case K: The active period of m starts during the last scan phase before “OFF” state and ends during the first scan phase after “OFF” state. Let $\delta \in [0, S]$ represent the difference between the start of the last scanning phase before “OFF” 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 $y - L_i^{off} - (S - \delta)$. Thus, the probability that p_i receives m in the second scanning phase is $\frac{y - L_i^{off} - (S - \delta)}{S} = \frac{y - L_i^{off} + \delta}{S} - 1$. Combing these two scanning phases, therefore, the probability that p_i can receive m (in at least one of the scanning phases) is

$$1 - (1 - (1 - \frac{\delta}{S})) \cdot (1 - (\frac{y - L_i^{off} + \delta}{S} - 1)) \\ = 1 - \frac{\delta}{S} \cdot (2 - \frac{y - L_i^{off} + \delta}{S})$$

Cases L~N: As shown on Fig 10, case L~N overlap at least one complete scan phase with m 's active period. Therefore, the reception probabilities in these cases are 1.

Fig. 11. Minimum update interval for $N_{scan} = N_{idle}$

Case O: The active period of m starts in the last idle phase before “OFF” state and ends in the following “OFF” state. Since both idle phase and “OFF” state cannot receive message from others, the reception probability for p_i to receive m is 0.

Case P: The active period of m starts and ends within the “OFF” period. Similar to the previous case, the reception probability is 0 for p_i to receive message m .

Case Q: The active period of m starts during the last idle phase before “OFF” state and ends during the first scan phase after “OFF”. Let $\delta \in [0, S]$ be the overlap of m 's active period and the first scan phase after “OFF”. Then, the reception probability for p_i to receive m is $\frac{\delta}{S}$.

Case R: The active period of m start within the last scan phase before “OFF” and ends in the first scan phase after “OFF”. This case is similar with case F and K except the interval between two scans is $L_i^{idle} + L_i^{OFF}$. Therefore, the reception probability for p_i to receive m is,

$$1 - (1 - (1 - \frac{\delta}{S})) \cdot (1 - (\frac{y - L_i^{off} - L_i^{idle} + \delta}{S} - 1)) \\ = 1 - \frac{\delta}{S} \cdot (2 - \frac{y - L_i^{off} - L_i^{idle} + \delta}{S})$$

Case S: The active period of m starts during the last scan phase and ends within the following “OFF” state. Let $\delta \in [0, S]$ be the difference between the beginning of the scan phase and that of m ’s active period. The reception probability for p_i to receive m is $1 - \frac{\delta}{S}$.

Case T: The active period of m starts during the last idle phase before “OFF” and ends during the first idle phase after “OFF” state. This active period completely contains the first scan phase after “OFF”. Thus, the reception probability for p_i to receive m is 1.

VII. IMPLEMENTATION AND EVALUATION

In this section, we will first introduce the system implementation of our PASA system and then present the performance evaluation results from both smartphones experiments and simulation.

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. 12 and Fig. 13 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. 12 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.

In addition, we develop a simulator to evaluate a system with a large number of smartphones. For both experiments and simulation, we set the length of a scanning phase to be 10 seconds, i.e., $S = 10$.

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 $\sum_i t_i$).

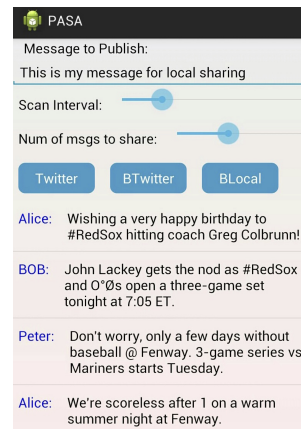


Fig. 12. Main page

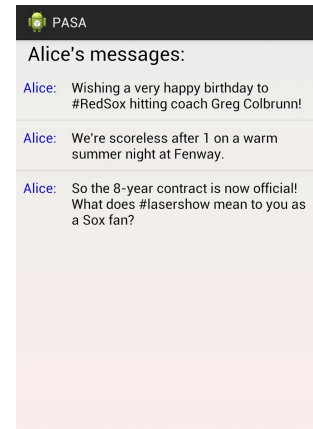


Fig. 13. Message list

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 participated 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. 14a 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. 14a. 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. Another observation from Fig. 14a is that when α is small, the average complete time of each phone varies a lot. This is because the expected message reception probability is low which indicates that it is likely to miss such a message during the first stage.

Fig. 14b 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. 14c and Fig. 14d 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.

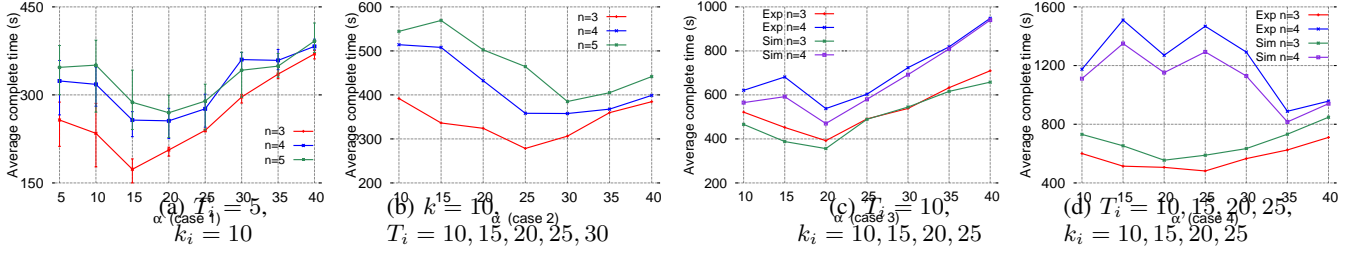
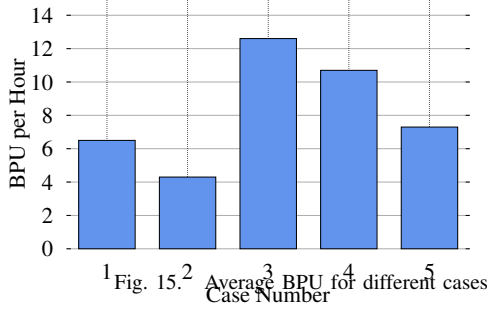
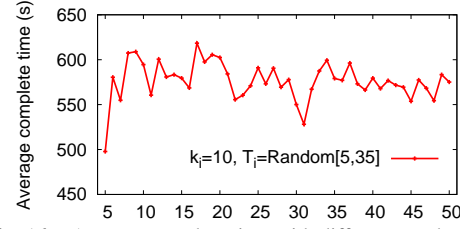
Fig. 14. Average complete time with different α (experiment)

Fig. 15. Average BPU for different cases

In Fig. 14c, $T_i = 10$ 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 14d, 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. 14d very well. The battery life is an important metric that directly affect the relief and victims. We measure the power consumption in terms of Battery Percentage Usage (BPU). Five different cases have been tested on our testbed for five times each, PASA with 10-second idle(1), PASA with 10-second idle and 10-second OFF state(2), WiFi Browsing(3), WiFi Direct transmission(4), Bluetooth transmission(5). The Fig. 15 shows the average cost of BPU of under different cases. As we can see from the figure, PASA with OFF state outperforms other experiments. The reason is that, in PASA-Off, the active time is 33% less than PASA.

2) *Simulation Results:* To evaluate a large scale network, we conduct simulation to test the performance of PASA. Our goal is to study the impact of the number of phones on the system performance. Fig. 16 shows the average complete time with $k_i = 10$ and $T_i = \text{Random}[5, 35]$ 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

Fig. 16. Average complete time with different numbers of phones (n), while $k_i = 10$ for all smartphones and $T_i = \text{Random}[5, 35]$

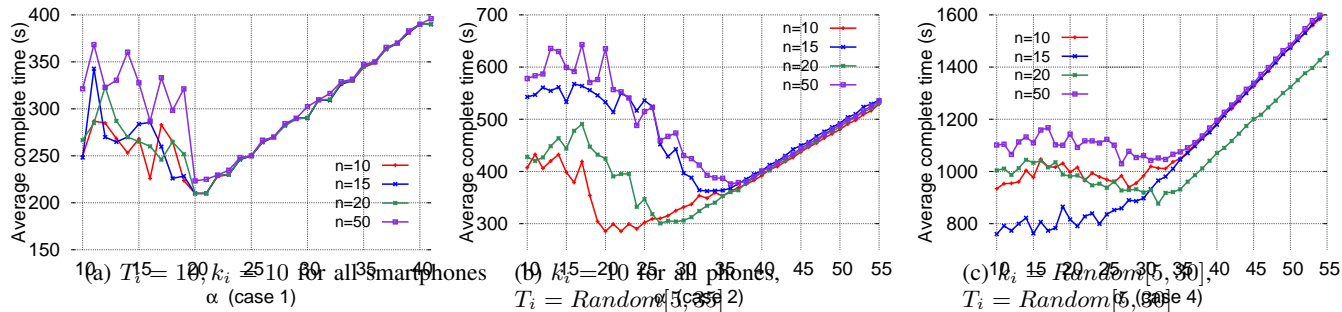
of total messages increase. This is caused by the many-to-one communication during each scan. Increasing the number of phones does not increase the average complete time much which indicates passive broadcast is certainly a scalable model.

Fig. 17a 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. 17a, our model can yield a good value of α for PASA system in these settings.

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

Fig. 17b shows the result when $k = 10$ for everyone and each phone randomly choose a 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 value of those four settings are 20, 26, 25, 33. Finally, Fig. 17c shows a general scenario where a user's k_i and T_i are randomly chosen from a given range, 5 to 30 in our setting. 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.

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

Fig. 17. Average complete time of PASA with different α (simulation)

parameter setting with Fig. 17c and the results are illustrated in Fig. 18. Note that we set a upper bound of 10,000 seconds for the execution time in our simulation and the missing values in Fig. 18 indicate that the simulation for those settings exceeds the upper bound of the execution time. Comparing to Fig. 17c, we observe that our two-stage PASA system significantly reduce 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.

Next, we present the results for PASA with OFF state that can benefit the disaster victims with low battery percentages. In these simulations, the smartphones randomly choose an OFF state length within the range of 0 to 20 seconds. Fig. 19a and 19b illustrate the performance for case 1 and 4. Comparing with the same cases in Fig 17, we have two observations. First, PASA-Off's the average complete time is much longer than PASA. By introducing the OFF state, not only reduces the the total length of active duration in each smartphone, but also significantly decreases reception probabilities. Second, for consecutive α , the average complete time jitters. This is because the smartphone randomly picks up a length of OFF state that results in a large difference of the update intervals of the second stage. The theoretical optimal value of those four settings of PASA-Off in Fig 19a are 26, 37, 34, 28 which, we can see, are at the bottom.

VIII. DISCUSSIONS AND FUTURE WORK

In this section, we discuss some limitations of our solution and possible extensions and future work.

First, our solution can be extended to a more general multi-stage protocol to further improve the performance. The two-stage solution in this paper is an extreme case where the second stage use the most conservative message update interval that is reasonable for post-disaster communication. An advanced approach would be a multi-stage solution, where the update interval is dynamic across the stages.

In addition, the problem setting can be relaxed to assume a general connected network topology where messages are spread in a multi-hop fashion. The problem objective would be adjusted accordingly and the analysis is expected to be more complicated. But our work in this paper can certainly serve as a starting point for this problem.

IX. CONCLUSION

This paper develops PASA, a local message dissemination system based on *passive broadcast* model. We present a two-stage protocol for propagate 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 system is effective and efficient for disseminating local messages.

REFERENCES

- [1] X. Chen, L. Cheng, C. Liu, Q. Liu, J. Liu, Y. Mao, and J. Murphy, "A woa-based optimization approach for task scheduling in cloud computing systems," *IEEE Systems Journal*, 2020.
- [2] Y. Mao, J. Oak, A. Pompili, D. Beer, T. Han, and P. Hu, "Draps: Dynamic and resource-aware placement scheme for docker containers in a heterogeneous cluster," in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2017, pp. 1–8.
- [3] Y. Mao, Y. Fu, S. Gu, S. Vhaduri, L. Cheng, and Q. Liu, "Resource management schemes for cloud-native platforms with computing containers of docker and kubernetes," 2020.
- [4] H. H. Harvey, Y. Mao, Y. Hou, and B. Sheng, "Edos: Edge assisted offloading system for mobile devices," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–9.
- [5] A. Acharya, Y. Hou, Y. Mao, and J. Yuan, "Edge-assisted image processing with joint optimization of responding and placement strategy," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2019, pp. 1241–1248.
- [6] A. Acharya, Y. Hou, Y. Mao, M. Xian, and J. Yuan, "Workload-aware task placement in edge-assisted human re-identification," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–9.
- [7] "Facebook Places," <http://www.facebook.com>.
- [8] "Twitter," <https://twitter.com/?lang=en>.
- [9] "Google Map," <https://www.google.com/maps>.
- [10] "Waze," <http://www.waze.com>.
- [11] "Siri," <http://www.apple.com/ios/siri/>.
- [12] "Cortana," <https://www.microsoft.com/en-us/mobile/experiences/cortana/>.
- [13] W. Zheng, M. Tynes, H. Gorelick, Y. Mao, L. Cheng, and Y. Hou, "Flowcon: Elastic flow configuration for containerized deep learning applications," in *Proceedings of the 48th International Conference on Parallel Processing*, 2019, pp. 1–10.
- [14] Y. Fu, S. Zhang, J. Terrero, Y. Mao, G. Liu, S. Li, and D. Tao, "Progress-based container scheduling for short-lived applications in a kubernetes cluster," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 278–287.
- [15] W. Zheng, Y. Song, Z. Guo, Y. Cui, S. Gu, Y. Mao, and L. Cheng, "Target-based resource allocation for deep learning applications in a multi-tenancy system," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2019, pp. 1–7.

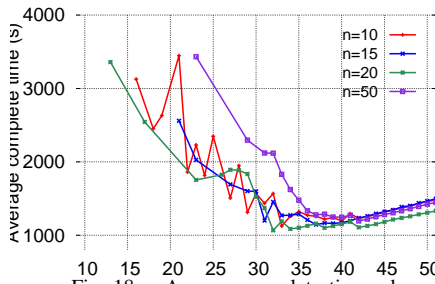


Fig. 18. Average complete time when messages are chosen with a simple rotation algorithm (in comparison to Fig. 17c)

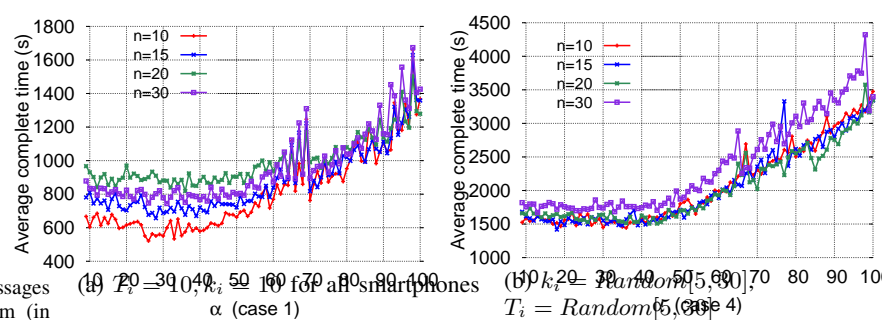


Fig. 19. Average complete time of PASA-Off with different α (simulation)

- [16] "Smartphone daily usage," <http://time.com/4147614/smartphone-usage-2015/>.
- [17] Y. Mao, J. Wang, J. P. Cohen, and B. Sheng, "Pasa: Passive broadcast for smartphone ad-hoc networks," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2014, pp. 1–8.
- [18] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opportunistic networks," in *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '08, 2008.
- [19] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun, "Geocommunity-based broadcasting for data dissemination in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, Apr. 2013.
- [20] B. K. Polat, P. Sachdeva, M. H. Ammar, and E. W. Zegura, "Message ferries as generalized dominating sets in intermittently connected mobile networks," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, ser. MobiOpp '10, 2010, pp. 22–31.
- [21] Y. Mao, J. Wang, and B. Sheng, "Mobile message board: Location-based message dissemination in wireless ad-hoc networks," in *2016 international conference on computing, networking and communications (ICNC)*. IEEE, 2016, pp. 1–5.
- [22] C. Peng, G. Shen, Y. Zhang, and S. Lu, "Point&Connect: intention-based device pairing for mobile phone users," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09, 2009.
- [23] B. Dodson, I. Vo, T. Purtell, A. Cannon, and M. Lam, "Musubi: disintermediated interactive social feeds for mobile devices," in *Proceedings of the 21st international conference on World Wide Web*, ser. WWW '12, 2012, pp. 211–220.
- [24] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, Nov. 2011.
- [25] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "Wifi-opp: ad-hoc-less opportunistic networking," in *Proceedings of the 6th ACM workshop on Challenged networks*. ACM, 2011, pp. 37–42.
- [26] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.
- [27] Y. Zhang and J. Zhao, "Social network analysis on data diffusion in delay tolerant networks," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '09. New York, NY, USA: ACM, 2009, pp. 345–346. [Online]. Available: <http://doi.acm.org/10.1145/1530748.1530806>
- [28] O. Turkes, H. Scholten, and P. J. M. Havinga, "Opportunistic beacon networks: Information dissemination via wireless network identifiers," in *The 5th IEEE International Workshop on the Impact of Human Mobility on Pervasive Systems and Applications (PerMoby 2016)*, in conjunction with IEEE International Conference on Pervasive Computing, PerCom 2016.
- [29] —, "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. New York, NY, USA: ACM, 2015, pp. 25–30. [Online]. Available: <http://doi.acm.org/10.1145/2799371.2799382>
- [30] A. Moghadam, S. Srinivasan, and H. 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.
- [31] X. Wu, M. Mazurowski, Z. Chen, and N. Meratnia, "Emergency message dissemination system for smartphones during natural disasters," in *ITS Telecommunications (ITST), 2011 11th International Conference on*. IEEE, 2011, pp. 258–263.
- [32] S. M. George, W. Zhou, H. Chenji, M. Won, Y. O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, no. 3, pp. 128–136, 2010.
- [33] D. Jaliha, R. Koilpillai, P. Khawas, S. Sampooram, S. H. Nagarajan, S. Kalpalatha, R. Bhavani, K. Takeda, and K. Kataoka, "A stand-alone quickly-deployable communication system for effective post-disaster management," in *Humanitarian Technology Conference (R10-HTC), 2014 IEEE Region 10*. IEEE, 2014, pp. 52–57.
- [34] Z. Lu, G. Cao, and T. La Porta, "Networking smartphones for disaster recovery," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.
- [35] S. Jakubczak, "Dythr," <http://szym.net/dythr/>.
- [36] T.-Y. Huang, K.-K. Yap, B. Dodson, M. S. Lam, and N. McKeown, "PhoneNet: a phone-to-phone network for group communication within an administrative domain," in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, ser. MobiHeld '10, 2010.
- [37] —, "Phonenet: a phone-to-phone network for group communication within an administrative domain," in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*. ACM, 2010, pp. 27–32.
- [38] A. C. Champion, Z. Yang, B. Zhang, J. Dai, D. Xuan, and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, Aug. 2013.