Mobile Peer-to-peer Data Dissemination with Resource Constraints[1]

Ouri Wolfson Bo Xu R. Michael Tanner
Department of Computer Science, University of Illinois at Chicago
{wolfson,boxu,rmtanner}@uic.edu

Abstract

Peer-to-peer data dissemination in a mobile ad-hoc environment is characterized by three resource constraints, including energy, communication bandwidth, and storage. Most of the existing studies deal with these constraints separately. In this paper we propose an algorithm called RANk-based DIssemination (RANDI), which provides an integral treatment to the three constraints. The contribution is in determining how to prioritize the reports in terms of their relevance, when to transmit the reports, and how many to transmit. We experimentally compare RANDI with 7DS and PeopleNet, two mobile peer-to-peer dissemination algorithms. The results show that RANDI significantly outperforms both algorithms.

1. Introduction

Mobile Peer-to-Peer (MP2P) (or gossiping, or epidemic) data dissemination is a paradigm in which a set of mobile devices (PDA's, vehicles, sensors) communicate with each other via unregulated, short-range wireless technologies such as IEEE 802.11 or Bluetooth. Each mobile device may produce data items (from now on called *reports*) and it may also be interested in receiving certain reports. Since the ranges of 802.11 or Bluetooth are not sufficient to reach all interested mobile devices, the dissemination is done by transitive multi-hop transmission. For transitive dissemination the intermediate devices (also called *brokers*) need to save reports and later, as new neighbors are discovered, transfer these reports. Thus each mobile device in the network is a broker, and additionally it may be a consumer or a producer of reports, or both. Observe that MP2P encompasses both MANET's (where the network is mobile but generally connected) and DTN's (where the network is mobile and subject to connectivity disruptions).

An important application domain of MP2P dissemination is matchmaking in social networks and mobile e-commerce ([5][3]). For example, in a large professional, political, or social gathering, the technology is useful to automatically facilitate a face-to-face meeting based on matching profiles (represented as reports). MP2P can also be used to propagate the detection of victims in a disaster recovery mission; or in general to relay alerts and sensed information in a mobile sensor network. An example of a report in the latter case is a data item indicating the availability of a parking slot at a certain location, at a certain time.

In contrast to a centralized database system, an MP2P system is not guaranteed to deliver all the matching reports to each consumer. The objective of an MP2P report dissemination system is to maximize the number and timeliness of matching reports delivered to the average consumer. This capability clearly depends on the amounts of device-resources allocated to the dissemination task. In this paper we study MP2P dissemination in an environment where the mobile devices are characterized by three resource constraints, including energy, communication bandwidth, and storage. We say that these devices are *bandwidth-energy-storage constrained* or *BES-constrained*. Mobile phones, PDA's, and low-cost/low-power sensors are BES-constrained. Vehicles are not energy-constrained because the energy needed for computation and wireless communication is negligible compared to the energy that is produced by a tank of fuel. In this paper we develop a MP2P method that disseminates reports to as many interested mobile devices and with as short a delay as possible, under the BES constraints.

An enormous number of studies have been conducted on data dissemination in mobile ad-hoc environments (see [6] for a good survey), and many of these studies address resource constraints. However, most of them deal with one or two aspects of BES but not all the three. For example, SPIN [7] and 7DS [2] deal with power conservation and (to some extent) bandwidth utilization, but provide no autonomous strategies for storage management. PeopleNet [3] deals with storage management and bandwidth utilization but does not touch power management. In these studies the resource conservation strategies are developed as independent algorithms and it is not clear how to combine them in order to consider the three resource types. Furthermore, each of these studies deals with either high mobility where a mobile device frequently encounters new neighbors (as in a vehicular network) or low mobility where the neighborhood is relatively stable (as in a sensor network). Our method works for both mobility types. Finally, many of these studies do not distinguish between the participation roles of a mobile device as a broker, producer, or consumer, which we feel is important for optimization.

---

[1] Research supported by NASA Award NNA06AA25C, and NSF Grants OII-0611017, DGE-0549489, 0513736, 0326284, and CCR-0231099.

On the surface, the impacts of the three (B/E/S) constraints on the design of data dissemination algorithms seem to be consistent, in the sense that a dissemination strategy that efficiently utilizes one resource (e.g., energy) is supposed to automatically efficiently utilize the other two (e.g., bandwidth and storage). However, this is not necessarily true. For example, the SPIN protocol uses meta-data negotiation before initiating the real data operation to minimize the redundant data transmission, hence, save energy over classical flooding. However, the meta-data exchange is based on broadcasting, which incurs flooding storm problems and therefore is not efficient in bandwidth utilization (see [8]).

We believe that the BES constraints should receive an integral consideration. In particular, the following issues need to be addressed:

1. When should a mobile device initiate communication of reports? Should the communication be initiated when two devices encounter each other (as in PeopleNet), or when new reports are received (as in SPIN), or periodically (as in 7DS)?
2. To whom to communicate (unicast or broadcast, if unicast, which neighbor)?
3. How many reports to communicate so that the bandwidth and energy are best utilized? Observe that if mobile devices transmit too much, then many collisions would reduce the number of successfully received reports; and if they transmit too little, report dissemination would suffer.
4. What to communicate (i.e., How to prioritize the communication if the number of reports to be communicated exceeds the optimal communication volume)?
5. What to save, or what reports to remove when the storage overflows?

In this paper we propose an algorithm called *Rank Based Dissemination* (RANDI) that addresses all the above issues. To the best of our knowledge, this is the first algorithm that deals with the BES constraints in an integral manner. Specifically, the RANDI algorithm includes the following novel techniques:

1. A two-phase protocol for the peer-to-peer interaction between two encountered mobile devices (i.e., the devices that come into the transmission range of each other). In the first phase, the encountered devices exchange queries and receive answers. This phase satisfies the interacting devices as report consumers. In the second phase, they exchange reports that enhance each other's capability as a broker.
2. A peer-to-peer interaction is a combination of one-to-one and broadcast communication, and it is triggered by either the discovery of a new neighbor or the reception of new reports. This paradigm reduces duplicate transmissions by disseminating only new reports to old neighbors and old reports only to new neighbors.
3. A strategy used by a mobile device to prioritize the reports based on their relevance. Intuitively, the relevance of a report depends on its hotness and size. Queries are disseminated to enable the estimation of hotness. When prioritizing for transmission, the strategy also considers whether the report is transferred to satisfy the receiver as a consumer or enhance its functionality as a broker.
4. A formula to compute the optimal transmission amount of each mobile device for each interaction. Using this formula a mobile device dynamically adjusts the transmission amount based on the length of the period of time between subsequent P2P interactions, such that overall energy efficiency is maximized while the energy budget constraint is satisfied.

Let us comment that in the RANDI algorithm, power conservation is achieved by saving the energy consumed by the wireless interface card for transmission. Another effective (and widely studied) paradigm is wake-up based power management (see e.g., [9]), which conserves power by saving the energy for *listening*. These two approaches are orthogonal in the sense that they control the different states (transmission and listening) of the wireless interface card and can be jointly used in a MP2P network.

The rest of the paper is organized as follows. Section 2 introduces the model. Section 3 describes the RANDI algorithm. Section 4 compares RANDI with 7DS and PeopleNet. Section 5 concludes the paper.

2. Model

2.1. Basics

Our system consists of a finite set of point (i.e. without an extent) *mobile devices.* During the period of time for which the system is studied, new devices may enter and existing devices may leave the system. Each device knows its neighbors (i.e., the devices within its transmission range) at any point in time, by using a neighbor-discovery protocol. Tracking neighbors enables a device to detect when encounters with new neighbors occur, which in turn trigger the execution of the RANDI algorithm. Occasionally, a mobile device O produces a *report* R having some unique *report-id,* and a size s(R). Each O also issues *queries* that express O's interests in certain types of information[2]. The queries issued by O are *native* to O.

Each O has a database, called the *reports database*, which stores the reports that O has produced or has received from other mobile devices. The size of the reports database is S_O bytes. In addition to reports, O also receives from its neighbors queries. O accumulates them in a *queries database* which stores the queries that O has issued or has received from other mobile devices. The size of the queries database is N_O bytes.

2.2. Energy Budget and Consumption Model

Before participating in reports dissemination, each mobile device user specifies the energy constraint as follows: "from now until time H the MP2P system is allowed to use fraction F of the remaining energy" (The rest is used for voice

---

[2] Note that O can be both a report producer and a report consumer at the same time.

communication, internet access, etc.). The allocated energy covers all the energy consumed by report dissemination, including the energy used for transmission, receiving, listening, and computation. F is called the *energy allocation fraction*. Given Ω Joules of remaining energy, this constraint is translated into the following specification: "The RANDI algorithm may use no more than $\Omega \cdot F$ Joules until time H". The pair ($\Omega \cdot F$, H) is the *energy budget*. Thus, we accommodate the lifetime demand of each individual device.

Now we introduce the energy consumption model. Let the size of a message be M bytes excluding the MAC header. According to [4], the energy consumed for transmitting a message can be described using a linear equation.

$$En=f\cdot M+g \qquad (1)$$

Intuitively, there is a fixed component associated with the network interface state changes and channel acquisition overhead, and an incremental component which is the size of the message. Experimental results confirm the accuracy of the linear model and are used to determine values for the linear coefficients g and f. For 802.11 broadcast, g=266×10^{-6} Joule, and f=5.27×10^{-6} *Joule*/*byte* (see [4])[3].

3. Rank-Based Dissemination Algorithm

3.1. Overview of the RANDI Algorithm

Intuitively, RANDI is an integration of multiple mechanisms that are aimed at disseminating reports to as many interested mobile devices and with as short a delay as possible, under the BES constraints. These mechanisms include:

1. **When to communicate.** The execution of RANDI consists of a sequence of send-and-receive operations. There are two types of operations. The first type is *query-response* (*QR*) mode, which is triggered when two mobile devices encounter each other. The second type is *relay*, which is triggered when a mobile device has new reports to disseminate. This *dual-type* mechanism makes RANDI automatically adapt to different mobility environments. In a highly dynamic[4] and/or partitionable environment, RANDI disseminates reports mainly via the encounters (QR mode); in a static environment (where there are rare encounters), RANDI disseminates reports mainly via proactive transmission of newly produced reports (relay mode).
2. **How much to communicate.** Observe that during a P2P operation a mobile device may have a lot of reports to transmit but it may not be able to transmit all of them due to bandwidth and energy constraints. How many reports a mobile device can transmit in a QR or relay operation is determined such that (i) the bandwidth/energy is best utilized; and (ii) the energy budget is uniformly consumed during the budgeted time period. At a high level, the transmission size is jointly determined by two factors. The first factor is the transmission size that optimizes the utilization of bandwidth and transmission energy. Intuitively, if the transmission size is too small, then the bandwidth is underutilized and the report dissemination suffers. In addition, the fixed overhead component f (see Eq. (1)) makes a small-size transmission less energy efficient. On the other hand, if the transmission size is too big, then many collisions would reduce the number of successfully received reports. Thus there is an *optimal transmission size* that achieves the best tradeoff between the bandwidth/energy utilization and transmission reliability.

   The second factor determining the transmission size is the amount of energy allocated to the operation. In other words, the mobile device may not be able to use the optimal transmission size due to the energy constraint. Specifically, during each second there is an amount of energy available for RANDI operations. The computation of this amount depends on the total allocation and consumption of energy so far, and is discussed in section 3.4. Based on this amount and the length of the time period from the last operation until the current operation, the energy allocated to the operation is computed; then the *maximum transmission size* supported by this amount of energy is computed (according to Eq. (1)). Then the final transmission size is the minimum between the optimal transmission size and the maximum transmission size. In other words, the final transmission size is the value that is closest to the optimal transmission size and is smaller than the maximum transmission size. For example, if the optimal transmission size is 10K bytes and the maximum transmission size is 5K, then the final transmission size is 5K.
3. **How and what to communicate.** A QR operation has two phases. In the first phase, the encountered mobile devices utilize the available energy to exchange their queries and receive answers. In the second phase, the rest of the available energy allocated to the exchange is used to exchange reports that enhance the other peer's capability as a broker, i.e. reports that are in high demand but do not satisfy the received query. The reports are transmitted by broadcast so that the other neighboring nodes may overhear the transmission, and thus their broker capability will also be enhanced. Thus, the QR operation is a combination of one-to-one and broadcast communication, and the RANDI algorithm is a combination of push and pull, in sense that the first phase of QR is pull, and "broker enhancement" and relay is push.

Now observe that since the amount of transmission is limited, not all the reports that satisfy the query or enhance the broker capability can be transmitted. Ranking is done to determine which reports to transmit. Second, since RANDI transmits a report either to satisfy the peer as a consumer (i.e. answer its query) or enhance its capability as a broker, there are two types of rank. A report has a *consumer-rank* when it is ranked to satisfy a device as a consumer, and a *broker-rank* when it is ranked to enhance a device's capability as a broker. The broker-rank is also used by the receiving peer to accommodate the most popular reports in the limited space of the reports database. The broker-rank of a report R at a device O depends on the following two factors.

a. Hotness, which represents the probability that R is queried by an arbitrary device in the network. This

---

[3] Measured with Lucent IEEE 802.11 2 Mbps WaveLAN PC card 2.4 GHz Direct Sequence Spread Spectrum.

[4] Observe that there can be two reasons for an environment to be dynamic. One is high mobility. Another is high turn-over, namely the mobile devices frequently enter and exit the system.

probability is estimated by the fraction of queries in *O*'s queries-database that are satisfied by *R*; the more queries *R* satisfies, the higher the rank of *R*. The queries database stores the latest queries that *O* received from encountered devices. In other words, *O* uses a sliding window of queries to determine the hotness of a report.

b. The size of *R*, denoted by s(*R*). The smaller s(*R*), the higher the rank of *R*; so to disseminate as many reports as possible.

Formally, the broker-rank of *R* is hotness(*R*)/s(*R*). The justification to this formula is given in subsection 3.2.

The consumer-rank of *R* depends solely on *s*(*R*) but not on the hotness. This is so because as a consumer, *O* wants to receive as many answers as possible, and therefore a shorter answer always has a higher priority, regardless of its hotness[5]. Formally, the consumer-rank of *R* is 1/s(*R*) (or any other monotonically decreasing function of s(*R*)).

4. **What to save**. Given the limited space of the reports database, a mobile device saves the reports that have the highest broker-ranks. In other words, we assume that the answers received by the mobile device are presented to the user, and possibly moved to the application area. Thus the reports saved in the reports database are solely for the purpose of brokering.

The rest of this section is organized as follows. In subsections 3.2 we formalize the problem of reports selection for brokering and justify the broker-rank formula. In subsections 3.3 and 3.4 we discuss how to determine the optimal transmission size and the maximum transmission size, respectively. In subsection 3.5 we describe formally the RANDI algorithm.

3.2. Reports Selection for Brokering

In this subsection we justify the broker rank of a report R defined above, i.e., hotness(*R*)/s(*R*). Let U be a set of reports stored at *O*, and T be the transmission size (when ranking to determine what to transmit) or the reports-database size (when ranking for determining what to save). When selecting reports out of U for the purpose of brokering, it is desirable that the selection includes as many answers to an arbitrary device encountered in the future as possible. Formally, the *reports selection for brokering (RSB) problem* is to construct a subset U' of U, such that the sum of the hotness values of the reports in U' is maximized, subject to the constraint that the sum of the sizes of the reports in U' does not exceed T. Intuitively, as far as *O* is concerned, U' includes more answers to an arbitrary device than any other subset of U that does not exceed the size limit T.

The RSB problem is straightforwardly transformed to the Knapsack problem and therefore is NP-complete (see [11]). In this paper we take an approximation solution to the RSB problem as follows. Order the set U=$\{R_1, R_2, \ldots, R_n\}$ by broker-rank, so that hotness(R_1)/s(R_1)≥hotness(R_2)/s(R_2)≥…≥hotness(R_n)/s(R_n). Starting with U' empty, proceed sequentially through this list, each time adding R_i to U' whenever the sum of the sizes of the reports already in U' does not exceed T-s(R_i). Then, compare the total hotness of U' to the hotness of the solution consisting solely of the hottest report whose size is smaller than T, and take the better of the two. We refer to this algorithm as *Greedy RSB* (or GRSB). According to [11], the absolute performance ratio for GRSB is 2. That is to say, for any instance of the problem, the total hotness of the solution produced by GRSB is at least half of the optimal solution. This approximation justifies the broker-rank formula.

3.3. The Optimal Transmission Size

Consider a broadcast by a mobile device *x*. In this subsection we develop and optimize a formula giving the efficiency of the broadcast per unit of energy consumed by the broadcast. In subsection 3.3.1 we give the formal definition of throughput and energy efficiency. In 3.3.2 we present an analytical model for optimization of energy efficiency.

3.3.1. Definitions of Throughput and Energy Efficiency.

We target mobile devices that use a carrier-sense multiple access (CSMA) protocol, e.g. 802.11. In such a network time is divided into slots, mobile devices communicate by broadcasts, and each broadcast lasts an integral number of time slots. For example, the length of the 802.11b time slot is 20µs.

Consider a broadcast of *M* bytes by a mobile device *x*. If another neighbor of *y* transmits during some time slot of the broadcast, then a *collision* occurs, and the whole broadcast is considered corrupt (i.e. unsuccessfully received) at *y*. Let *F* be the number of neighbors that successfully receive the message from *x*. The *throughput of the broadcast by x*, denoted *Th*, is defined to be: $Th = M \cdot F$. Intuitively, the throughput is the total amount of data successfully received by neighbors of *x*.

Let *En* be the energy consumed at the network interface of *x* for sending the broadcast message. The *energy efficiency* of the broadcast by *x*, denoted *PE*, is defined to be: $PE = \frac{Th}{En}$. In other words, the energy efficiency is the throughput produced by each unit of transmission energy consumed at *x*.

TABLE I. Summary of symbols used in computing the energy efficiency.

Symbol	Meaning
λ	Number of devices per unit of the geographic area (we assume uniform spatial distribution).
r	Transmission range of each device in meters.
b	Data transmission speed in bits per second.
Th	Throughput of a broadcast.
En	Transmission energy consumed by the broadcast
M	Size of each broadcast in bytes.
p'	Probability that a device starts a broadcast at an arbitrary medium access time slot.

---

[5] We assume that a report either satisfies a query or it does not, i.e., the degree of satisfaction is either 0 or 1 but nothing in between.

τ	Length of the medium access time slot in seconds.
h	Size of Medium Access Control header in bytes.
c	Number of seconds since the completion of the last broadcast until the start of the current broadcast

3.3.2. Analytical Model of Energy Efficiency. First, let us present a formula for the computation of the throughput *Th* which is introduced in [5]. Let a mobile device x execute a broadcast at an arbitrary time slot. Under the assumptions and notations given in Table I, *Th,* the throughput of the broadcast is a random variable. According to [5], the expected value of *Th* can be approximated by

$$E(Th) \approx 2 \cdot \pi \cdot \lambda \cdot M \cdot \int_0^r \delta \cdot (1-p')^{\lambda \cdot r^2 \cdot (2 \cdot q(\frac{\delta}{2 \cdot r}) + (\pi - 2 \cdot q(\frac{\delta}{2 \cdot r})) \cdot (2T+1)) - 1} d\delta \qquad (2)$$

where $T=(M+h)\cdot 8/b \cdot \tau$, and for a fraction a, $q(a)=\arccos(a)-a\sqrt{1-(a)^2}$. Eq. (2) takes into consideration the effect of hidden terminals as well as direct collisions.

By the definition of energy efficiency, the expected value of the energy efficiency is

$$E(PE) \approx 2 \cdot \pi \cdot \lambda \cdot M \cdot \frac{\int_0^r \delta \cdot (1-p')^{\lambda \cdot r^2 \cdot (2 \cdot q(\frac{\delta}{2 \cdot r}) + (\pi - 2 \cdot q(\frac{\delta}{2 \cdot r})) \cdot (2T+1)) - 1} d\delta}{f \cdot M + g} \qquad (3)$$

Now consider Eq. (3). If τ, p', λ, h, b, r, f, and g are fixed, then the energy efficiency *PE* as a function of the broadcast size M is a bell curve. Thus there is a value of M that maximizes the energy efficiency, i.e. achieves the best tradeoff between the channel utilization and broadcast reliability. And this value is computed and used by the RANDI algorithm given in subsection 3.5.

For the rest of this subsection we show that indeed, except for M, all the parameters of Eq. (3) can be determined by a mobile device. The parameters τ, h, r, b depend on the network, and are fixed for a given communication network technology. For example, h is 47 in 802.11b. f and g depend on the network interface hardware and can be calibrated a priori as demonstrated by [4]. The density λ can be estimated based on the average number of neighbors over time (recall that each mobile device knows its neighbors via the neighbor-discovery protocol), given the transmission range.

The probability p' is determined as follows. Let c be the number of seconds since the completion of the last broadcast of O until the time when the current broadcast size is to be determined. If a mobile device starts a broadcast every c seconds on average, then its probability of starting a broadcast in each medium access time slot is τ/c. Thus we substitute the broadcast probability p' in Equation (3) by τ/c. For example, if c=5 seconds and τ=20μs, then $p'=(20\times10^{-6})/5=4\times10^{-6}$.

Using $p'=\tau/c$ [6] we obtain a formula for the throughput in which the only unknown is M. Thus we can find the *optimal* M, i.e. the value of M for which *PE* is maximized. Denote this value by $M_{optimal}$.

3.4. The Maximum Transmission Size

In order to compute the maximum transmission size, before executing an operation RANDI computes the amount of energy that it consumed from its initialization until now, and determines how much energy remains available for it. The energy consumed for transmitting and receiving is tracked as follows. For each operation executed by RANDI (e.g., a transmission or a reception of reports), the algorithm reads the amounts of the battery energy before and after the operation. The difference between the two is the energy consumed by the operation. The energy for listening is tracked differently. Since listening services all the short range wireless applications including RANDI, the energy cost of listening is split among the applications according to some formula. For example, if there are 5 wireless applications, then each application is charged 1/5 of the listening energy. Thus at any point in time the device is able to compute Ω_{avail} the remaining energy that is available for RANDI until time H (see sec. 2.2). Specifically, let $\Omega_{consumed}$ be the total amount of energy that has been consumed by RANDI. Then $\Omega_{avail}=\Omega \cdot F-\Omega_{consumed}$.

Let T_{last} be the length of the time period from the end of the last broadcast until time H. The maximum size of the current broadcast, denoted M_{max}, is computed as follows:

$$M_{\max} = \frac{c \cdot (\Omega_{avail}/T_{last}) - g}{f} \qquad (4)$$

where c is defined in Table I and f, g are defined in Eq. (1). Intuitively, Ω_{avail}/T_{last} is the amount of energy allocated for RANDI per second from the end of the last broadcast until time H. $c \cdot (\Omega_{avail}/T_{last})$ is the energy accumulated from the end of the last broadcast until now. Thus M_{max} computed by Eq. (4) is the maximum broadcast size supported by the accumulated energy.

3.5. Description of the Algorithm

The QR operation is executed at a mobile device A when A encounters a new neighbor. The relay operation is executed by A at a fixed time interval after the latest broadcast[7]. Now we describe the QR operation and the relay operation executed at A, respectively.

---

[6] Observe that the actual p' may be lower than τ/c. This is because there may be a delay from the time when the broadcast is triggered until the channel is sensed free and the broadcast is actually started. In other words, the actual broadcast period may be bigger than c. However, since the difference between p and p' is small, this delay is expected to be small as well and therefore can be ignored.

[7] Assume that the time-interval is 3 minutes. If no broadcast is executed within the last 3 minutes, then relay is initiated.

Query-response operation (*A* encounters a device *B*)

Let Q_A and Q_B be the native queries of *A* and *B* respectively. Let IDS_A be the set of the id's of the reports in *A*'s reports database, and IDS_B be the set of the id's of the reports in *B*'s reports database.

1. *A* sends Q_A and IDS_A to *B* by unicast (Figure 1).
2. *A* receives Q_B and IDS_A-IDS_B from *B*.
3. *A* puts Q_B in its queries database (FIFO-maintained).
4. *A* computes M=min($M_{optimal}$, M_{max}) where $M_{optimal}$ and M_{max} are computed as discussed in subsections 3.3 and 3.4 respectively.
5. *A* fills up a message of *M* bytes in the following order:
   a. the reports in *A*'s reports database that satisfy Q_B and their id's are in IDS_A-IDS_B (these are the answers to Q_B that are unknown to B). If all the reports in this category do not fit in the message, they are picked up in the order of their consumer-ranks (see subsection 3.1).
   b. other reports in *A*'s reports database whose id's are contained in IDS_A-IDS_B (these are the broker-enhancement reports). If all the reports in this category do not fit in the message, then the GRSB algorithm described in subsection 3.2 is executed to select the reports to include in the message.
6. *A* broadcasts the *M*-bytes message.
7. Symmetrically, *A* receives reports from *B* and puts them in its reports database. If the size of the reports database is bigger than S_A (Recall that S_A is the size limit of *A*'s reports database), then the GRSB algorithm is executed to select the reports for saving.

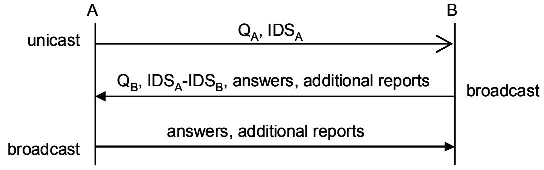


Figure 1. Communication in the QR operation

If the QR operation does not complete before *B* moves out of the transmission range of *A*, then the operation is stopped. No rollback is needed. If *A* encounters another device *C* during its QR operation with *B*, the QR operation with *C* will be suppressed until the QR operation with *B* finishes. The QR operation with *C* is canceled if the distance between *A* and *C* is greater than the transmission range after the QR operation with *B* finishes.

Relay operation

Let X be the set of reports in *A*'s reports database that have not been previously transmitted by *A*. *A* computes M_A, the number of bytes in the current broadcast, using the formula in step 4 above (QR.4). *A* fills up a message of M_A bytes with the reports in X using the GRSB algorithm, and broadcasts the message. However, the broadcast is suppressed if *A* does not have any neighbor.

Observe that both QR and relay operations are totally distributed in the sense that they do not rely on any dedicated device to support or coordinate the operations. Thus the failures of individual devices do not necessarily disable the running of the MP2P network. The failures simply result in the decrease of device density.

4. Comparison with 7DS and PeopleNet

4.1. The 7DS and PeopleNet Algorithms

For 7DS we compare with its P-P scheme[8] ([2]). In the 7DS P-P scheme, each mobile device *O* periodically broadcasts its native query. When receiving the query, each neighbor searches its cache and broadcasts the reports that match the query. *O* caches the received reports so that it may later on answer queries from other mobile devices. *O* stops participating in the system if the energy allocated for reports transmissions is used up. Since 7DS does not provide a cache-purging strategy, we randomly chose reports to remove when the size of the cache exceeded the memory allocated to the reports database. For each parameter configuration, we tested 7DS with the inter-query intervals 10, 15, and 50 to 900 seconds with increments of 50 seconds. 7DS with inter-query interval *T* seconds is referred to as 7DS-*T*.

In PeopleNet a P2P interaction occurs when two mobile devices encounter each other (see [3]). During the interaction, the encountering devices exchange part of their reports databases, and randomly pick reports to purge so that the size limit of the reports database is accommodated. PeopleNet assumes that *L* reports can be transmitted by a mobile device at each interaction, but it does not provide a method to determine the value of *L*. In our implementation device A transmits to device B all the reports in A's database that are unknown to B, and vice versa.

The main differences of 7DS and PeopleNet from RANDI are: 1) there is no energy management for determining the transmission size; 2) the broker function is much more simplistic (no ranking); 3) 7DS does not have a good strategy to determine when to communicate.

4.2. Simulation Environment

The three algorithms are implemented in SWANS (Scalable Wireless Ad-hoc Network Simulator) built at Cornell University. We augmented SWANS with a feature that tracks the energy consumed by 802.11 for transmitting and receiving each message. The listening energy is assumed to be zero for all, RANDI, 7DS, and PeopleNet. In the simulations, we used 802.11b with the data transmission speed of 2M bits per seconds and the transmission range of 100 meters. *N* devices move within a 400meter×400meter square area according to the *random way-point* mobility model with mean speed 1 mile/hour and mean pause time 180 seconds. The whole simulation runs for 3600 seconds. Each device has a life span which follows a normal distribution with the mean of 900 seconds and the standard deviation of 300 seconds. When the life span of a device expires, it is removed from the system and a new device is created. Thus

---

[8] 7DS has another scheme called Server/Client which is not applicable in our context.

the number of *live* mobile devices is fixed. Each time unit is 1 second.

For representing reports and queries, we adopted the Number Intervals (NI) subscription model introduced in [1]. Particularly, a report is represented by a point within the real interval [0, 1]. A query is represented by a range within [0, 1], e.g., [0.2, 0.7]. A report R matches a query Q if R's number falls into Q's range.

Reports are produced by a Poisson process with intensity u. Each report's number is randomly chosen from the [0, 1] interval. An arbitrary live mobile device becomes the producer of the report R.

Each mobile device has a native query which is generated when the device is introduced to the system, and is fixed for the life span of the device. The range of the query is generated by choosing a center and a length. The length of the range is selected randomly according to a normal distribution with mean 0.05 and variance 0.002. The query-center falls into the [0, 1] interval following a Zipf distribution. In particular, the [0, 1] interval is divided into 10 disjoint sections ([0, 0.1), [0.1, 0.2), …). The probability that a query-center falls into the i-th ($1 \le i \le 10$) section is $(1/i)/\sum_{j=1}^{10}(1/j)$. In other words, the resources are uniformly distributed, and the queries are distributed according to Zipf's law.

The size of each report is randomly chosen from [100, 2000] bytes. The size of each query is fixed to be 100 bytes. The size limit of each reports database is randomly chosen from [$0.5 \times S$, $1.5 \times S$] bytes where S is fixed to be 100K. In the simulations with the 7DS and PeopleNet, we expanded the reports databases of each mobile device to match the queries database overhead in RANDI.

The energy budget is initialized for each mobile device as follows. We assumed that each mobile device user expects the battery to last for 8 hours. With 15984 joules of the total battery energy (1200mAh, 3.7V)[9], we computed the energy available per second for report dissemination to be $0.555 \times F$ Joule (Recall that F is the energy allocation which is a parameter of the simulation system). Based on this we computed the energy budget for each mobile device during its life span. For example, if the life span of the device is 900 seconds, and F is 0.01, then the energy budget for the device is $0.555 \times 900 \times 0.01 = 5$ Joule, for all the three algorithms.

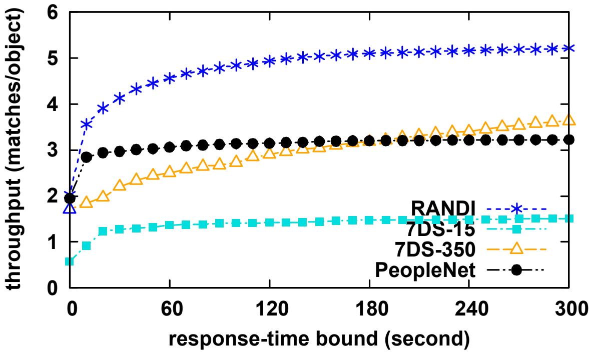
4.3. Performance Measure

The performance measure considers the average number of matches received by a mobile device with response-times smaller than a certain time limit w. The response-time of a report R received at a mobile device O is defined as follows. The response time starts at the time at which O is introduced or R is introduced, whichever is later since both must be present, and ending when O receives R. This measure is called the *response-time bounded throughput*, or *throughput*, and c is called the *response-time bound*. By varying the value of w, we evaluate the throughput of an algorithm under different response-time constraints. This is similar to the way an academic department is evaluated according to the percentage of its students that graduate within 4 years, 5 years, etc.

4.4. Simulation Results

Figure 2 shows the response-time bounded throughput of the three algorithms for a parameter configuration. We conducted experiments with different bandwidth and memory allocations. The results are similar, but for space considerations we need to omit these. For 7DS we display two curves. One curve is 7DS-15 [10] since for most experiments conducted in [3] the query-interval is 15 seconds. Another 7DS curve is the 7DS variant that receives most matches regardless of the response time. It turns out that the curve that generates the most matches differs for different parameter configurations; in Fig. 2 this curve is 7DS-350. Clearly, this way of comparison is unrealistically optimistic for 7DS, because in real life a mobile device does not know the system parameters such as the average reports database size and the report production rate, and therefore it cannot determine the best query-interval to use in 7DS.

Figure 2. energy allocation(F)=0.01,
reports database size(S)=100Kbytes
inter-device distance=40 meters, 0.1 report/second(u)



From Fig. 2 it can be seen that RANDI significantly outperforms both 7DS and PeopleNet. The advantage of RANDI over the two algorithms depends on the parameter configuration and the response-time bound. For example, in Fig. 2, when the response-time bound is 300 seconds, the throughput of RANDI is 3 times more than that of 7DS-15 and 1/3 more than those of 7DS-350 and PeopleNet. Thus, compared to 7DS-15, RANDI receives 3 times more reports that are younger than 300 seconds. The advantage of RANDI is due to the following reasons.

1. In 7DS and PeopleNet, there is no control of the transmission size to optimize the utilization of energy and bandwidth. For example, in Fig. 2, with 7DS-15 and PeopleNet, the energy budget is used up within the first quarter of the life span of an average mobile device. Furthermore, 7DS-15 is too aggressive in bandwidth

---

[9] Specification of the battery for iPAQ HW6500.

[10] Recall that for each parameter configuration 7DS has variants 7DS-10, 7DS-15,…,7DS-900 where 7DS-T is 7DS with inter-query interval of T seconds.

consumption (with frequent transmissions), and generates excessive collisions. RANDI, on the other hand, controls the transmission size to obtain the best tradeoff between bandwidth/energy utilization and broadcast reliability, and to uniformly schedule the energy consumption throughout the budgeted area.

2. RANDI has a better "transmission-triggering" mechanism. In 7DS and PeopleNet, a report transmission is triggered either by a time-event (in 7DS) or by an encounter (in PeopleNet). With the time-event trigger, the event frequency (i.e., query-interval) is hard to determine as explained earlier. With the encounter trigger, the report dissemination suffers when the mobility is low and when the report production rate is high. RANDI combines the time-event trigger and the encounter trigger and therefore accommodates various mobility and report production environments.

3. In 7DS there are duplications among the answer sets returned by different neighbors to the same query and among the consecutive answer sets returned by the same neighbor. In other words, the communication efficiency is very low. RANDI, on the other hand, strives to reduce duplicate transmissions by disseminating only new reports to old neighbors and old reports only to new neighbors.

4. In 7DS and PeopleNet reports are randomly purged from the reports database, without considering their sizes and the system-wide demand to them. RANDI prioritizes reports for transmitting and saving based on their hotness and sizes.

Let us comment now on the impact of the communication reliability to the performance of RANDI. In other words, how robust RANDI is as the 802.11 channel quality deteriorates, due to, for example, the interference of Bluetooth communication [10]. For this purpose we artificially introduced environmental noise to the SWANS system and tested RANDI under a wide range of bit error rates (from 10^{-9} to 10^{-3}). The experiments show that the performance of RANDI is rather stable when the bit error rate is lower than 10^{-5}, and it drops drastically when the bit error rate is higher than 10^{-5}. Furthermore, we evaluated the benefit of the forward error correction (FEC) scheme to RANDI. In a FEC scheme, the sender adds redundant data to its messages, which allows the receiver to detect and correct errors (within some bound) without the need to ask the sender for additional data. The experiments show that the benefit of a FEC scheme depends on its error correcting capability, the channel quality, and the coding rate (i.e., the ratio between the payload length and the message length). For example, with a FEC scheme like the one proposed in [14], when the bit error rate is 10^{-4}, the performance of RANDI is improved by 3 times compared to the case where FEC is not used.

5. Conclusion

In this paper we proposed a distributed algorithm called RANDI for MP2P dissemination with resource constraints. RANDI achieves efficient utilization of the bandwidth/energy/storage by a comprehensive solution to the decision issues including when to communicate, how (broadcast or unicast) and how much to communicate, and what to communicate and save in the local database. Experimental results show that RANDI significantly outperforms existing methods.

Future work includes extensions in several directions. First, it is interesting how to take advantage of knowledge that reports have different reliability-factors. For example, how to treat a report that is generated by a semi-reliable source? The second research direction is transmitting to smaller distances to reduce power consumption. In other words, transmissions of the same device at different times may have different ranges. This type of work was carried out for static networks (e.g. [12]). Another research direction is mechanisms to incentivize devices to act as brokers. Recent work (see e.g., [13]) provides a good starting point for this research.

References

[1] Y. Huang, H. Garcia-Molina. Publish/Subscribe Tree Construction in Wireless Ad-Hoc Networks. *MDM'03*.

[2] M. Papadopouli and H. Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. *MobiHoc*, 2001.

[3] M. Motani, et al. PeopleNet: Engineering A Wireless Virtual Social Network. *MobiCom*, 2005.

[4] L. Feeney, M. Nilson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. *INFOCOM*, 2001.

[5] Ouri Wolfson, et al. Search-and-Discover in Mobile P2P Network Databases. ICDCS, 2006.

[6] A. Datta, et al. Autonomous Gossiping: A self-organizing epidemic alg. for selective information dissemination in wireless mobile ad-hoc networks. ICSNW'04.

[7] J. Kulik, et al. Adaptive protocols for information dissemination in wireless sensor networks. MobiCom'99.

[8] M. Medidi, et al. Data dissemination using gossiping in wireless sensor networks. SPIE'05.

[9] R. Zheng, et al. Asynchronous wakeup for ad hoc networks. MobiHoc 2003.

[10] R. Punnoose, et al. Experimental Results for Interference between Bluetooth and IEEE 802.11b DSSS Systems. IEEE Vehicular Society Conference 2001.

[11] M. Garey and D. Johnson, *Computer & Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[12] P. Santi. The Critical Transmitting Range for Connectivity in Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 4(3), 2005.

[13] A. Mondal, S. Madria, M. Kitsuregawa. ConQuer: A Peer Group-based Incentive Model for Constraint Querying in Mobile-P2P Networks. MDM 2007.

[14] J. Chen and R. M. Tanner, "A Hybrid Coding Scheme for the Gilbert-Elliott Channel," IEEE Trans. Comm., Vol. 54, No. 10, October, 2006, pp. 1787-1796.