

Contact Duration Aware Data Replication in Delay Tolerant Networks

Xuejun Zhuo*, Qinghua Li†, Wei Gao†, Guohong Cao†, Yiqi Dai*

* Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, China. Email: zhuoxj07@mails.tsinghua.edu.cn; dyq@mail.tsinghua.edu.cn

†The Pennsylvania State University, USA. Email: {qxll118, weigao, gcao}@cse.psu.edu

Abstract—The recent popularization of hand-held mobile devices, such as smartphones, enables the inter-connectivity among mobile users without the support of Internet infrastructure. When mobile users move and contact each other opportunistically, they form a Delay Tolerant Network (DTN), which can be exploited to share data among them. Data replication is one of the common techniques for such data sharing. However, the unstable network topology and limited contact duration in DTNs make it difficult to directly apply traditional data replication schemes. Although there are a few existing studies on data replication in DTNs, they generally ignore the contact duration limits. In this paper, we recognize the deficiency of existing data replication schemes which treat the complete data item as the replication unit, and propose to replicate data at the packet level. We analytically formulate the contact duration aware data replication problem and give a centralized solution to better utilize the limited storage buffers and the contact opportunities. We further propose a practical contact Duration Aware Replication Algorithm (DARA) which operates in a fully distributed manner and reduces the computational complexity. Extensive simulations on both synthetic and realistic traces show that our distributed scheme achieves close-to-optimal performance, and outperforms other existing replication schemes.

I. INTRODUCTION

Due to the recent popularization of hand-held mobile devices, such as smartphones, there arises the requirement to effectively distribute data to those devices. Such data can generally be distributed from the service provider to mobile users via cellular networks. However, it has been recently reported that the excessive traffic demands are overloading the cellular network infrastructure [1]. To address this problem, some recent studies [2], [3] have proposed to utilize the mobility and the subsequent opportunistic contacts of the users to offload part of the cellular traffic, especially for the bandwidth-eager traffic, such as video clips. Particularly, the mobile devices with short-range wireless interfaces can form a Delay Tolerant Network (DTN) [4] by exploiting their peer-to-peer opportunistic connectivity. A mobile user downloads and replicates data from the service provider when it has a low-cost connection to the Access Points (APs), such as WiFi hotspots, and then distributes data to other peer users when they contact each other via DTNs.

This work was supported in part by the US National Science Foundation (NSF) under grant number CNS-0721479, and by Network Science CTA under grant W911NF-09-2-0053.

Data replication has been widely used to improve the performance of data access in traditional wired/wireless networks [5], [6], [7]. With data replication, users can access the data without the support of network infrastructure, and can reduce the traffic load of the infrastructure. The fundamental question in data replication is: how to determine the optimal replication placement to better utilize the limited storage space and the transmission bandwidth. However, existing data replication techniques cannot be directly applied to DTNs which are characterized by unstable network topology and limited contact duration.

The challenge of unstable network topology has been well studied in data forwarding in DTNs, and addressed by exploiting node mobility model [8] or using social network theory [9], [10]. Existing works on data replication in DTNs [11], [12] address this challenge by modeling the contact processes as a Poisson distribution according to the contact histories. However, these works ignore the contact durations limits. They simply assume that the complete data can always be transmitted as long as a requester contacts a node storing the data. In other words, they consider a data item as integral during data replication. A node either replicates the complete data, or does not replicate it at all. This methodology is referred to as *data-level replication*.

Unfortunately, in reality, the contact duration is usually short due to node movement and the limited range of peer-to-peer wireless communication. For example, when hand-held devices communicate via Bluetooth, which supports a typical wireless range of about 10 meters, the contact duration tends to be as short as several seconds if the users are moving at a walking speed. When the users are in the high speed vehicles, even if they communicate via WiFi which has a longer range, the contact duration is still short. In addition, the transmission of large multimedia content, such as video, further exacerbates the impact of the limited contact duration on data replication.

In this paper, we identify the deficiency of traditional data-level replication in realistic DTN environments where the contact duration may be short. To better utilize the network resource, we adopt the erasure coding technique [13] to encode a data item into multiple coded packets, and propose *packet-level replication* for DTNs. We focus on appropriately determining which data items and how many packets to replicate at each node based on node mobility pattern and data access pattern.

We further validate the advantages of packet-level replication by analytically proving that a node provides a diminishing contribution gain to other nodes when more packets are replicated. Thus, our scheme achieves a better cost-effectiveness than replicating the complete data on a node. To the best of our knowledge, this is the first paper to identify the impact of contact duration on data replication in DTNs. The contributions of the paper are three-fold:

- We formulate the contact duration aware packet-level replication problem to determine the optimal replication solution, which maximizes the average data retrieval probability, given the storage space constraint and data retrieval time constraint.
- We formulate the replication problem as a Mixed Integer Programming (MIP) problem, and give a centralized solution using the CPLEX optimization suite [14].
- We design a distributed replication scheme based on a novel *replication benefit* metric. Extensive synthetic and trace-driven simulations validate that our solution achieves close-to-optimal performance, and outperforms other existing replication schemes.

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents an overview of the network model and the basic idea of our design. Section IV formulates the contact duration aware data replication problem in DTNs, and Section V provides a distributed replication algorithm. The results on both synthetic and trace-driven performance evaluation are presented in Section VI, and Section VII concludes the paper.

II. RELATED WORK

To increase the performance of data access in DTNs, many existing works focus on the topic of data dissemination. In [15], a broadcasting based data dissemination approach is implemented. In [16], the authors provide theoretical analysis to the stationary and transient regimes of data dissemination. Some later solutions disseminate data based on a pub/sub structure, in which the data is classified into some pre-defined channels, and disseminated based on user's subscriptions [17].

Recently, caching solutions have been proposed to improve the performance of data access in DTNs. For example, Gao *et al.* [18] propose to intentionally cache data at a set of network central locations which can be easily accessed by other nodes, but the contact duration limitation has not been taken into account. Zhuo *et al.* [19] considers the effects of contact duration on caching, but the analysis is limited to a given fixed number of replicas for a data item.

Data replication is another solution to improve data access, which has been well studied in unstructured peer-to-peer systems [7], [20]. In [7], Cohen *et al.* prove that the square-root allocation strategy can minimize the expected search size on successful queries. In [20], Tewari *et al.* show that if the nodes use an expanding ring search, the proportional allocation strategy can lead to optimal performance. Data replication problem becomes even tough under heterogeneous network environments. The problem of determining the optimal replication solution in a heterogeneous network is similar to

the facility location problem and the K-median problem, and both of them are proved to be NP-hard. A 20.5-approximation algorithm for the data replication problem with uniform-size data items in heterogeneous networks has been proposed in [6]. Later, Tang *et al.* [5] design a polynomial-time centralized replication algorithm which can achieve a 4-approximation solution, and a localized distributed algorithm in heterogeneous ad hoc networks. However, these data replication schemes cannot be applied to DTNs.

In [11], a distributed data replication scheme has been proposed for DTNs which considers the impatience of the nodes towards the query delay of different data types. This scheme is based on the assumption of homogeneity, i.e., all nodes in the network have the same preferences and mobility pattern. Later, Loannidis *et al.* [12] formulate the data replication problem in heterogeneous DTNs, and design a fully distributed replication algorithm. However, these works do not consider the contact duration limits, and simply assume that as long as a data requester contacts a replicating node, the complete requested data can be retrieved.

III. OVERVIEW

In this section, we introduce the network model and the basic idea of our approach.

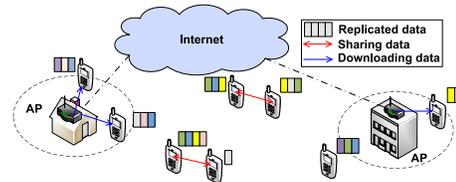


Fig. 1. Network scenario

A. Network Model

We consider a hybrid network scenario which includes APs and mobile nodes as shown in Fig. 1. The APs can connect to the data provider via the Internet. Mobile nodes download and replicate data when they move into the wireless range of the AP. We do not consider the contact duration limits between mobile nodes and APs, since the APs provide a relatively large coverage and the nodes are relatively static around APs, especially those at home or office. However, when mobile nodes are out of the AP's coverage, they can only share data with other peers. Thus, the mobile nodes form a DTN, for example via Bluetooth, to exchange their replicated data by opportunistic contacts. The amount of data that can be transmitted between two nodes is decided by their contact pattern such as contact frequency and contact duration.

The contact frequency has been well analyzed. Similar to [9], [21], we model the contact processes between nodes as the independent Poisson processes, and this modeling has been experimentally validated in [9], [22]. However, the contact duration has been overly simplified in most prior works. The assumptions on contact duration in DTNs, typically fall into two extremes: one assumes short contact duration where only one packet can be sent during a contact [23], [24], and the

other assumes long contact duration where an arbitrary number of packets can be sent during a contact [9], [25]. The protocols based on these assumptions lead to either over-pessimistic or over-optimistic results, since some studies [26], [27] have validated that the contact duration in DTNs follows a Pareto distribution. Based on [26], [27], we model the contact duration between nodes as a Pareto distribution.

Due to the contact duration limits, a complete data may not be transmitted. The simple fragmentation may result in the coupon collector's problem [28] which significantly decreases the efficiency of data access. To mitigate this problem, we adopt the erasure coding technique [13] to encode data into a large set of coded packets, and any sufficiently large subset of the packets can be used to reconstruct the data. Specifically, the coding process takes the original data of size G and a coding rate $\frac{1}{r}$ as the inputs, and outputs Gr/g uniformly sized coded packet of size g . Any $s = (1 + \epsilon)G/g$ coded packets can be used to reconstruct the data, where ϵ is a small constant which is determined by the exact erasure coding algorithm used.

B. Basic Idea

Our goal is to determine the replication solution for each node to fully utilize the network resources, such as the limited storage spaces that each node is willing to provide, and the limited node contact opportunities. Existing studies on data replication [5], [7], [11], [12] assume that the requester can always get the complete requested data from the replicating node, when a connection arises between them. They replicate data at data-level, i.e., a node either replicates the complete data or does not replicate any packet. However, in DTNs, when the contact duration is short, a complete data item may not be fully transmitted to the requester. Next, we use a simple example to show the deficiency of employing the data-level replication in the contact duration limited scenario.

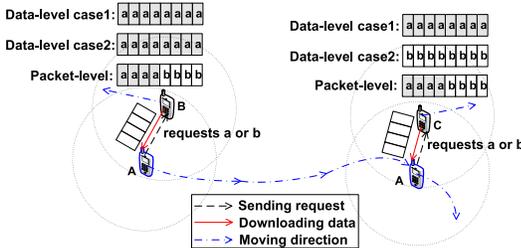


Fig. 2. An example of contact scenario

As shown in Fig. 2, node A is a data requester, and it sequentially contacts node B and C when it moves. During each contact, only four packets can be transmitted due to the contact duration limits. There are two data items named a and b , and both of them have eight packets. Both node B and C are only willing to provide limited buffers to replicate eight packets. If data-level replication is conducted, B and C can only select either a or b to replicate. If they replicate the same data, say a for example, then A can successfully download the data if it requests a , but it cannot download any packet if it requests b . If B and C replicate different data, e.g., B replicating a and C replicating b , then no matter which data

A requests, it can only download four packets of its requested data, which are not enough to reconstruct the original data. As can be seen, under the data-level replication, both the storage spaces and the contact opportunities have not been fully utilized.

To make better use of the resources by considering the contact duration limitation, we propose to replicate data at the packet-level. As shown in Fig. 2, if node B and C replicate four distinct packets of both data item a and b respectively, node A can successfully download the complete data (four packets from B and four packets from C), no matter which data it requests. We can see that given the same network resource, packet-level replication outperforms data-level replication. Therefore, in this paper, we design data replication schemes at the packet-level. By adopting erasure coding, all coded packets of a data item can be treated equally. Then, we only need to consider how many packets, rather than exactly which packets, of a data item to replicate at each node.

IV. PROBLEM FORMULATION AND ANALYSIS

In this section, we formulate the contact duration aware data replication problem and give a centralized solution.

A. Problem Definition

Suppose that there are n mobile nodes and m data items in the network, and each node, say i , is willing to provide a storage of size ρ_i for data replication. To simplify the presentation, we assume that all data items have the same size G , and each of them can be reconstructed by any $s = (1 + \epsilon)G/g$ coded packets. We also assume that the data items have the same retrieval time constraint T . We will discuss how to remove the two assumptions in Section IV-D. Let matrix \mathbf{x} represent the data replication solution, where each element $x_{i,d} \in \mathbf{x}$ denotes the number of packets of data d replicated at node i . Let variable $A_i^d(\mathbf{x})$ denote the total number of packets of data d that node i can retrieve from others within the time constraint, given a data replication solution \mathbf{x} . We aim to maximize the average successful data retrieval probability within the time constraint. We assume that the average inter-contact time between the mobile node and the AP is usually longer than the time constraint; otherwise, it is not attractive for the node to download data via DTN. Then, we have:

Definition 1: The contact duration aware data replication problem is to determine the optimal data replication solution \mathbf{x} to maximize the average data retrieval probability, subject to the storage constraint and the data retrieval time constraint.

$$\max \sum_{d=1}^m \sum_{i=1}^n q_{i,d} P(A_i^d(\mathbf{x}) \geq s - x_{i,d}) \quad (1)$$

$$s.t. \quad \forall x_{i,d} \in \mathbf{x}, x_{i,d} \in \{0, \dots, s\}, \quad (2)$$

$$\forall i \in \{1, \dots, n\}, \sum_{d=1}^m x_{i,d} \leq \rho_i. \quad (3)$$

where $q_{i,d}$ is the query rate of node i to data item d , and $\sum_{i=1}^n \sum_{d=1}^m q_{i,d} = 1$. $q_{i,d}$ is decided by the data access pattern of the network. Constraint (2) ensures that every node replicates

at most s packets for each data. Constraint (3) guarantees that the total number of packets replicated at each node is limited by the storage constraint of the node.

Though Definition 1 is straightforward, to get the optimal solution is quite complex. The main difficulty lies in deriving the closed form expression of the objective function. We know that $P(A_i^d(\mathbf{x}) \geq s - x_{i,d}) = \sum_{a=s-x_{i,d}}^{\infty} f_{A_i^d(\mathbf{x})}(a)$, where $f_{A_i^d(\mathbf{x})}(a)$ is the Probability Mass Function (PMF) of $A_i^d(\mathbf{x})$. To obtain the closed form expression of the objective function, we try to calculate $f_{A_i^d(\mathbf{x})}(a)$.

For any node pair, say i and j , we set a random variable $Y_{i,j}$ which follows Pareto distribution to represent the maximum amount of data that can be sent during a contact between them. We further define a random variable $Z_{i,j}$ which follows Poisson distribution to denote the number of contacts happened between them within the time constraint. Then, $Y_{i,j}^{(1)}, Y_{i,j}^{(2)}, \dots, Y_{i,j}^{(Z_{i,j})}$ are $Z_{i,j}$ i.i.d. variables which denote the maximum amount of data that can be sent during each contact between node i and j . The total amount of data that can be sent between them within the time constraint is denoted as $U_{i,j}$, and $U_{i,j} = Y_{i,j}^{(1)} + Y_{i,j}^{(2)} + \dots + Y_{i,j}^{(Z_{i,j})}$.

Let $A_{i,j}^d(\mathbf{x})$ denote the number of the coded packets of data item d that node i can receive from node j within the time constraint given a replication solution \mathbf{x} . The PMF of $A_{i,j}^d(\mathbf{x})$ can be calculated as:

$$f_{A_{i,j}^d(\mathbf{x})}(a) = \begin{cases} \int_{ga}^{g(a+1)} f_{U_{i,j}}(u) du & 0 \leq a < x_{j,d} \\ \int_{ga}^{\infty} f_{U_{i,j}}(u) du & a = x_{j,d} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

where $f_{U_{i,j}}(u)$ is the Probability Density Function (PDF) of $U_{i,j}$. The PMF of $A_i^d(\mathbf{x}) = \sum_{j \neq i} A_{i,j}^d(\mathbf{x})$ can be derived as:

$$f_{A_i^d(\mathbf{x})}(a) = f_{\sum_{j \neq i} A_{i,j}^d(\mathbf{x})}(a) = f_{A_{i,1}^d(\mathbf{x})}(a) \otimes \dots \otimes f_{A_{i,i-1}^d(\mathbf{x})}(a) \otimes f_{A_{i,i+1}^d(\mathbf{x})}(a) \otimes \dots \otimes f_{A_{i,n}^d(\mathbf{x})}(a) \quad (5)$$

where $f_{A_i^d(\mathbf{x})}(a)$ is a discrete convolution of $f_{A_{i,j}^d(\mathbf{x})}(a)$, which can be derived from the PDF of $U_{i,j}$. However, since $U_{i,j}$ is the sum of a random number of random variables, its PDF has no closed form expression. As a result, we cannot derive the closed form expression of the objective function.

Next, to make the problem tractable, we isolate the variables in the objective function, and let the part which has no closed form only contain constants. Then, we give an approximate calculation on the part. We formulate the problem into a Mixed Integer Programming (MIP) and solve it by using the CPLEX.

B. MIP Formulation

Let random variable $V_{i,j}$ denote the maximum number of packets that can be transmitted between node i and j within the time constraint. $V_{i,j}$ can be calculated as follows:

$$V_{i,j} = \begin{cases} \lfloor \frac{U_{i,j}}{g} \rfloor & \lfloor \frac{U_{i,j}}{g} \rfloor < s \\ s & \lfloor \frac{U_{i,j}}{g} \rfloor \geq s. \end{cases} \quad (6)$$

Each $V_{i,j}$ has $s+1$ possible integer values ranging from 0 to s . Specially, $V_{i,j} = s$ when the aggregated contact duration

between the two nodes is long enough to transmit the complete data item. Therefore, the total number of packets that can be transmitted between node i and any other nodes form a $1 \times n$ vector $[V_{i,1}, V_{i,2}, \dots, V_{i,n}]$. This vector falls into one of the $(s+1)^n$ possible combinations. We use a $1 \times n$ vector $\mathbf{v} = [v_1, v_2, \dots, v_n]$ to denote one possible combination or say, a contact pattern in which v_j ($0 \leq v_j \leq s$, for $j = 1, \dots, n$) packets can be transmitted from or to node j . There are totally $(s+1)^n$ possible patterns \mathbf{v} , and we define Φ as the set of total possible patterns. Let $P_i^{\mathbf{v}}$ denote the probability that node i follows the contact pattern $\mathbf{v} = [v_1, v_2, \dots, v_n]$, i.e., $V_{i,j} = v_j$ for $j = 1, \dots, n$. Due to the independent contact processes of node pairs, $P_i^{\mathbf{v}} = \prod_{j=1}^n P(V_{i,j} = v_j)$. Note that for all i , $P(V_{i,i} = s) = 1$ and $P(V_{i,i} \neq s) = 0$. We will provide details on how to calculate $P_i^{\mathbf{v}}$ in Section IV-C.

We further define a binary variable $R_d^{\mathbf{v}}(\mathbf{x})$ to denote whether the contact pattern \mathbf{v} enables a node to retrieve enough packets to reconstruct data item d within the time constraint, given a replication solution \mathbf{x} . If the total number of packets of data item d that can be retrieved is equal to or larger than s , the data item can be reconstructed. Thus, $R_d^{\mathbf{v}}(\mathbf{x})$ can be calculated as below:

$$R_d^{\mathbf{v}}(\mathbf{x}) = \begin{cases} 1 & \sum_{j=1}^n \min(v_j, x_{j,d}) \geq s \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Given notations in Table I, the contact duration aware data replication problem in DTNs can be re-defined as follows:

$$\max \sum_{d \in \mathcal{M}} \sum_{i \in \mathcal{N}} \sum_{\mathbf{v} \in \Phi} q_{i,d} P_i^{\mathbf{v}} R_d^{\mathbf{v}}(\mathbf{x}) \quad (8)$$

$$\text{s.t. } \forall i \in \mathcal{N}, \forall d \in \mathcal{M}, x_{i,d} \in \{0, 1, \dots, s\}, \quad (9)$$

$$\forall i \in \mathcal{N}, \sum_{d \in \mathcal{M}} x_{i,d} \leq \rho_i, \quad (10)$$

$$\forall \mathbf{v} \in \Phi, \forall d \in \mathcal{M}, R_d^{\mathbf{v}} \in \{0, 1\}, \quad (11)$$

$$\forall \mathbf{v} \in \Phi, \forall d \in \mathcal{M}, \sum_{j \in \mathcal{N}} \min(v_j, x_{j,d}) \geq s R_d^{\mathbf{v}}(\mathbf{x}) \quad (12)$$

Since the objective function to maximize is monotonically increasing with variable $R_d^{\mathbf{v}}$, constraint (11) and (12) ensure that $R_d^{\mathbf{v}}(\mathbf{x})$ equals to 1 if and only if $\sum_{j \in \mathcal{N}} \min(v_j, x_{j,d}) \geq s$; otherwise, $R_d^{\mathbf{v}}(\mathbf{x})$ equals to 0. However, the *min* function in the last constraint makes the optimization problem nonlinear. Therefore, we replace this constraint with the following constraints by introducing a set of auxiliary variables $h_{j,d}^{\mathbf{v}}(\mathbf{x})$:

$$\forall \mathbf{v} \in \Phi, \forall d \in \mathcal{M}, \sum_{j \in \mathcal{N}} h_{j,d}^{\mathbf{v}}(\mathbf{x}) \geq s R_d^{\mathbf{v}}(\mathbf{x}), \quad (13)$$

$$\forall \mathbf{v} \in \Phi, \forall d \in \mathcal{M}, \forall j \in \mathcal{N}, h_{j,d}^{\mathbf{v}}(\mathbf{x}) \leq v_j, \quad (14)$$

$$\forall \mathbf{v} \in \Phi, \forall d \in \mathcal{M}, \forall j \in \mathcal{N}, h_{j,d}^{\mathbf{v}}(\mathbf{x}) \leq x_{j,d}. \quad (15)$$

where constraint (13), (14), and (15) are equivalent to constraint (12). In this way, the formulation turns to a MIP. It is easy to prove that this MIP formulation is equivalent to the formulation presented in Definition 1.

C. Calculation of P_i^v

P_i^v denotes the probability that node i follows the contact pattern $\mathbf{v} = [v_1, v_2, \dots, v_n]$. Thus, $P_i^v = \prod_{j \in \mathcal{N}} P(V_{i,j} = v_j)$, where $P(V_{i,j} = v_j)$ can be calculated as:

$$P(V_{i,j} = v_j) = \begin{cases} \int_{gv_j}^{g(v_j+1)} f_{U_{i,j}}(u) du & 0 \leq v_j < s \\ \int_{gv_j}^{\infty} f_{U_{i,j}}(u) du & v_j = s \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

However, the PDF of $U_{i,j}$ has no closed form expression. Next, we provide an approximate calculation on $P(V_{i,j} = v_j)$, by simplifying it under two cases according to the contact rate $\lambda_{i,j}$ between node i and j .

1) $\lambda_{i,j}T < 1$, which means that within time constraint T , the expected number of contacts between the two nodes is smaller than 1. Then, we simply assume that as long as the two nodes can contact within the time constraint, which has a probability $1 - e^{-\lambda_{i,j}T}$, they contact only once. Here, we do not consider the scenario of contacting multiple times due to its higher order probability. Under this simplification, the total amount of data transmitted between them follows the Pareto distribution, and $P(V_{i,j} = v_j)$ can be calculated as:

$$P(V_{i,j} = v_j) = \begin{cases} e^{-\lambda_{i,j}T} & v_j = 0 \\ (1 - e^{-\lambda_{i,j}T}) \int_{\max\{y_{i,j}, gv_j\}}^{g(v_j+1)} \frac{\tau_{i,j} y_{i,j}^{\tau_{i,j}}}{u^{\tau_{i,j}+1}} du & 0 < v_j < s \\ (1 - e^{-\lambda_{i,j}T}) \int_{\max\{y_{i,j}, gv_j\}}^{\infty} \frac{\tau_{i,j} y_{i,j}^{\tau_{i,j}}}{u^{\tau_{i,j}+1}} du & v_j = s \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

where $\tau_{i,j}$ and $y_{i,j}$ are the shape and scale parameter of $Y_{i,j}$, and $y_{i,j}$ represents the minimum possible value of $Y_{i,j}$. Note that the PDF of $Y_{i,j}$ is $f_{Y_{i,j}}(y) = \frac{\tau_{i,j} y_{i,j}^{\tau_{i,j}}}{u^{\tau_{i,j}+1}}$, when $y > y_{i,j}$.

2) $\lambda_{i,j}T \geq 1$, which means that nodes i and j are more likely to contact several times within the time constraint. We assume that the amount of data that can be transmitted during each contact equals to the expectation of $Y_{i,j}$. Under this simplification, $P(V_{i,j} = v_j)$ can be calculated as:

$$P(V_{i,j} = v_j) = \begin{cases} \frac{(\lambda_{i,j}T)^k e^{-\lambda_{i,j}T}}{k!} & v_j = \lfloor \frac{kE(Y_{i,j})}{g} \rfloor < s \\ 1 - \sum_{k=0}^{\lceil \frac{gs}{E(Y_{i,j})} \rceil - 1} \frac{(\lambda_{i,j}T)^k e^{-\lambda_{i,j}T}}{k!} & v_j = s \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

where k is a nonnegative integer.

D. Discussion on Nonuniform Data Size and Time Constraint

Although in previous sections, we have assumed that the data items have the same size and the same time constraint, our MIP formulation still works without these assumptions. Suppose that data item d can be reconstructed by any s_d coded packets of it, and its time constraint is T_d . Let s_{max} denote the minimum number of packets needed to reconstruct the data which has the maximum size among all data items. For any node pair i and j , $V_{i,j}$ has $s_{max} + 1$ possible integer values ranging from 0 to s_{max} . We replace the constant s in

TABLE I
NOTATION USED IN THE MIP FORMULATION

Indices:	
$\mathcal{N} = \{1, 2, \dots, n\}$	The set of nodes.
$\mathcal{M} = \{1, 2, \dots, m\}$	The set of data items.
Φ	The set of possible contact patterns.
Constants:	
$s > 0$	The number of coded packets needed to reconstruct a data item.
$\mathbf{v} = [v_1, v_2, \dots, v_n]$, $\forall v_j \in \{0, \dots, s\}$	One possible contact pattern.
$q_{id} \in [0, 1]$	The query rate of node i to data item d .
$\rho_i > 0$	The storage size in terms of coded packet that node i is willing to provide for data replication.
$P_i^v \in [0, 1]$	The probability that node i follows contact pattern \mathbf{v} .
Variables:	
$\mathbf{x}, \forall x_{id} \in \mathbf{x}$, $R_d^v(\mathbf{x}) \in \{0, 1\}$	Replication solution matrix. Indicator of whether contact pattern \mathbf{v} yields a successful data retrieval of data item d within the time constraint, given a replication solution \mathbf{x} .
$h_{jd}^v(\mathbf{x})$	Auxiliary variables for MIP formulation.

the MIP formulation with s_d , and replace P_i^v with $P_{i,d}^v$. The occurrence probability $P_{i,d}^v$ can be calculated using the same method presented in Section IV-C by replacing T with T_d . To simplify the presentation, we focus on the case that all data items have the same size and time constraint in this paper.

E. Problem Analysis

Traditional data-level replication problem can be reduced to the classical facility location problem, which has been proved to be NP-hard [5], [6]. Our packet-level replication problem is more complicated since it needs to determine not only what data items, but also how many packets of them to replicate at each node. We formulate the problem as a MIP which can be solved by the CPLEX optimization suite based on Branch and Bound. Branch and Bound algorithm solves optimization problems by implicit enumeration of the solution space, partitioning it into a search tree. Unfortunately, due to the integrity requirement, MIP problems are generally NP-hard, and in the worst case may require a search tree of exponential size. Our MIP problem is further complicated due to its exponential number of variables and constraints.

We use CPLEX to solve the problem in a small scale, and the results will be shown in Section VI as a benchmark. However, if the network is large, it is hard to derive the optimal solution due to the extremely high computational complexity. To make data replication more applicable to the practical use, we further design a polynomial time distributed algorithm based on the local knowledge of each node, without using global knowledge such as the query rate of each node to each data item, the contact rate between any node pair, and the current replication placement of the whole network.

V. DISTRIBUTED DATA REPLICATION ALGORITHM

In this section, we present our distributed Duration Aware Replication Algorithm (DARA).

A. Main Idea

DARA is a distributed algorithm run at each node. Based on node's local knowledge, DARA runs in a greedy manner, i.e., iteratively replicating the packet which brings the maximum replication benefit to other nodes, until the buffer is full. How to evaluate the replication benefit is the key to DARA. Intuitively, the benefit of replicating an additional packet of data item d into node i 's buffer is affected by three factors: 1) the popularity of data item d , 2) the capability of node i to contribute the packet to others, and 3) the current availability of data item d . If the data item has high popularity and the node has large capability to contribute the packet to others, the replication benefit is large. However, if data item d can be retrieved at a high probability without this replication, the replication benefit will be lower. Although some of these factors have been implicitly considered in traditional replication and DTN routing algorithms [5], [21], they should be addressed differently when the contact duration limits are considered, especially for 2) and 3).

To obtain the popularity of data items, each node maintains a data popularity table which records the average query rate to each data item from its local view. Each node counts the number of pending requests it has received (including those generated by itself), and calculates the average query rate to each data item d as $q_d = n_d/n_{total}$, where n_d is the number of requests for data item d , and n_{total} is the total number of requests. The node updates the data popularity table at each time window as $q_d = q_d^{old}\alpha + q_d^{new}(1 - \alpha)$, where q_d^{old} is the old query rate of data item d , and q_d^{new} is the new query rate derived in the latest time window. α is a decaying factor, which decides the weight of the old and new query rate.

Our goal is to maximize the average data retrieval probability. Thus, we use the retrieval probability to evaluate data availability. Each node maintains a data availability table which records the data retrieval probability of its contacted nodes to each data item. When two nodes contact, they exchange their data retrieval probability on each data item. To calculate the data retrieval probability, each node also maintains a data replication table, which records the replication placement of the contacted nodes. Based on the table, a node, say i , estimates its data retrieval probability to data item d as $P_{i,d} = \sum_{a=s-x_{i,d}}^{\infty} f_{A_i^d(\mathbf{x})}(a)$. $f_{A_i^d(\mathbf{x})}(a)$ can be derived via Eq. 5, and $f_{A_{i,j}^d(\mathbf{x})}(a)$ in Eq. 5 can be calculated as:

$$f_{A_{i,j}^d(\mathbf{x})}(a) = \begin{cases} P(V_{i,j} = a) & 0 \leq a < x_{j,d} \\ \sum_{v=x_{j,d}}^s P(V_{i,j} = v) & a = x_{j,d} \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

where $P(V_{i,j} = v)$ can be derived via Eq. 17, 18.

To evaluate the capability of a node to contribute a packet to others, we define a metric called *contribution gain*. Next, we describe how to calculate the contribution gain.

B. Contribution Gain

The contribution that node i can provide to j in terms of data d , denoted as $C_{i,j}^d(x_{i,d})$, is defined as the expected number of

packets of d that i can transmit to j within the time constraint.

$$\begin{aligned} C_{i,j}^d(x_{i,d}) &= E(A_{j,i}^d(x_{i,d})) \\ &= \sum_{v=1}^{x_{i,d}-1} vP(V_{i,j} = v) + \sum_{v=x_{i,d}}^s x_{i,d}P(V_{i,j} = v). \end{aligned} \quad (20)$$

The contribution gain represents the increment in the contribution provided by a node by replicating an additional packet, which reflects the capability of the node to contribute the newly replicated packet to others.

Definition 2: For node i which has replicated $x_{i,d}$ packets of data item d , the contribution gain that it can provide to node j in terms of data item d is defined as:

$$\begin{aligned} \Delta C_{i,j}^d(x_{i,d}) &= C_{i,j}^d(x_{i,d} + 1) - C_{i,j}^d(x_{i,d}) \\ &= 1 - \sum_{v=0}^{x_{i,d}} P(V_{i,j} = v). \end{aligned} \quad (21)$$

where $P(V_{i,j} = v)$ can be derived via Eq. 17, 18. Note that the contribution gain provided by a node to itself equals 1, since the node can directly enjoy its replicated data.

As can be seen, $\Delta C_{i,j}^d(x_{i,d})$ is a non-increasing function of $x_{i,d}$, which means that the contribution gain a node can provide is diminishing as more packets are replicated at this node. Intuitively, this phenomenon can be explained as: the more packets of a data item that a node replicates, the more contact opportunities needed to transmit all of them to the data requester. Since there is a lower probability when more contact opportunities happen, replicating more packets causes a lower contribution gain. This also implies that the data-level replication results in a low utilization of both the storage spaces and the contact opportunities. Compared to replicating a complete data in a node which brings lower contribution gain, it is more beneficial to split part of the data into other nodes. In this way, the total replication contribution is improved with the same storage cost, and more contact opportunities can be utilized. Therefore, the packet-level replication is more suitable for contact duration limited scenario, in which each node can adaptively decide how many packets to replicate for each data item, according to its contact patterns.

C. Replication Benefit

Based on the above analysis, we define the replication benefit provided by replicating an additional packet of data item d to node i 's buffer, which is determined by the popularity and availability of the data item, and the contribution gain that the node can provide to other nodes:

Definition 3: The replication benefit provided by replicating an additional packet of data item d at node i which has already replicated $x_{i,d}$ packets of d can be defined as:

$$B_{i,d}(x_{i,d}) = \begin{cases} \frac{q_d}{n} \sum_{j \in \mathcal{N}} (1 - P_{j,d}) \Delta C_{i,j}^d(x_{i,d}) & x_{i,d} < s \\ 0 & x_{i,d} = s. \end{cases}$$

where the replication benefit turns to zero if node i has already replicated s packets of data d . The derivation of q_d and $P_{j,d}$ are

discussed in Section V-A. The intuition behind Definition 3 is that the replication benefit provided by replicating a new packet of data d on node i should be the contribution gain (weighted by the popularity of d) that i can provide to other nodes if they currently cannot successfully retrieve d from the network.

D. The Protocol

In DARA, there are two cases for a node to download/replicate data: the node moves into the range of any AP or it contacts and swaps data with another node.

1) *Node-to-AP*: When a node moves into the wireless range of the AP, it iteratively replicates one packet of a chosen data item which can provide the maximum replication benefit, until no buffer space is left. More specifically, if node i has already replicated $x_{i,d}$ packets of data item d , the replication benefit of replicating an additional packet of the data item into the node is $B_{i,d}(x_{i,d})$. Node i downloads a packet of a chosen data item d_{\max} which has the maximum $B_{i,d_{\max}}(x_{i,d_{\max}})$ among all the data items, from the AP.

When the buffer is full, the node still iteratively replicates one packet using the same rule of data selection, namely a packet of a chosen data item d_{\max} with the maximum $B_{i,d_{\max}}(x_{i,d_{\max}})$ is downloaded. Meanwhile, to make room for the new packet, it needs to remove a packet that has the minimum replication benefit. Specifically, if node i has already replicated $x_{i,d}$ packets of data item d , removing any packet of the data will decrease the replication benefit by $B_{i,d}(x_{i,d} - 1)$. Thus the node removes a packet of a chosen data item d_{\min} with the minimum $B_{i,d_{\min}}(x_{i,d_{\min}} - 1)$. This process iterates until $B_{i,d_{\max}}(x_{i,d_{\max}})$ is not larger than $B_{i,d_{\min}}(x_{i,d_{\min}} - 1)$.

2) *Node-to-Node*: When two nodes contact, they can swap the data replicated in their buffer, and this data swap is controlled by the node with higher centrality value. Centrality metrics are widely used to measure the importance of the nodes in the network [9] [25]. However, the existing work only considers the contact frequency, and ignores the contact duration. In this paper, we use the expected number of packets which can be transmitted between the node to others within the time constraint as the centrality metric.

Definition 4: The centrality value of node i is defined as:

$$CEN_i = \frac{1}{n-1} \sum_{j \in \mathcal{N}, j \neq i} \sum_{v=1}^s vP(V_{i,j} = v). \quad (22)$$

The node with higher centrality is referred to as the *master*, and the node with lower centrality is referred to as the *slave*. The master downloads data from the slave according to the same rule used in the Node-to-AP case, until the connection is over or there is no suitable data to be downloaded from the slave. The only difference is that the master needs to send the packets that are replaced by the newly downloaded packets back to the slave. The slave replicates them, and removes the packets which have been sent to the master. Thus, no redundant coded packet is generated. The two nodes just swap some data to increase the replication benefit of the master. Note that although swapping data by considering the replication

benefit of both nodes is more beneficial, it requires more time-consuming information exchanges for coordination, which may not fit the contact duration limited scenario.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of DARA on both synthetic and realistic traces.

A. Schemes for Comparison

To evaluate the performance of DARA, we compare it with eight replication schemes: 1. **OPT**: The optimal solution derived from the MIP formulation using CPLEX; 2. **PSEPHOS**: An existing data-level replication scheme in DTNs [12]; 3. **UNI-data** (4. **UNI-packet**): A data-level (packet-level) replication scheme, where the storage space is evenly allocated among all data items; 5. **SQRT-data** (6. **SQRT-packet**): A data-level (packet-level) replication scheme, where the storage allocation is proportional to the square root of the query rate; 7. **PROP-data** (8. **PROP-packet**): A data-level (packet-level) replication scheme, where the storage allocation is proportional to the query rate.

In PSEPHOS, each node maintains a “vote” for each data item based on the information collected from others to rate the caching importance of the data item. Whenever a node meets the AP, it downloads the data with the top votes. However, this scheme ignores the contact duration limits, and replicates at data-level. The other six naive schemes are designed for comparison purpose. These schemes only consider the popularity of the data items, and replicate data items according to three typical replication rules: *uniform*, *square root*, and *proportional*. Under the same rule, the total number of buffers allocated to each data item remains unchanged between the data-level and packet-level schemes. The difference is that in data-level scheme, a node either replicates a complete data item, or does not replicate any packet of it, but in packet-level scheme, by adopting erasure coding each data item is more evenly allocated among the nodes, and every node only replicates parts of it.

B. Synthetic Traces

1) *Simulation Settings*: We generate a small-scale trace and a large-scale trace. In the small-scale trace (large-scale trace, resp.), there are 10 (100, resp.) mobile nodes and 10 (50, resp.) data items in the network. Each data item contains 8 (32, resp.) packets, and each node has storage of size 24 (96, resp.). In both traces, the contact rates between node pairs are randomly generated within the range of [0.0003, 0.005]. The contact duration is randomly generated following the Pareto distribution, where each node pair has its distinct shape and scale parameters of the distribution. The shape and scale parameters are within the range of [2, 4] and [1, 3] respectively, and then the expected number of packets that can be sent during a contact between a node pair is within the range of [1.33, 6]¹.

¹ $\frac{1}{\tau_{i,j} - 1} y_{i,j}$ is the expected number of packets that can be transmitted between node i and node j , where $\tau_{i,j}$ and $y_{i,j}$ are the shape and scale parameter.

To show the impact of contact duration, in the large-scale trace, we further generate a long contact duration scenario, where the shape and scale parameters are set within the range of $[1, 1.5]$ and $[5, 10]$ respectively, and hence the expected number of packets that can be transmitted during a contact between a node pair is within the range of $[15, \infty)$.

In both traces, we also add a static node to act as the AP, from which the mobile nodes can download and replicate data. The contact frequency between the AP and the mobile nodes is randomly generated within the range of $[0.0001, 0.001]$. We set the contact frequency low enough so that the mobile nodes prefer to retrieve data from other peers rather than wait for the contact opportunities with AP. We use the Zipf-like query distribution, where the query rate of the i^{th} most popular data item is proportional to i^{-w} , and w is set to 1 unless specified differently. We sort the data items in the decreasing order of its popularity, and number them from 1 to m . In each simulation run, the first $1/5$ of the trace is used for warmup. The presented results are averaged over 10 runs.

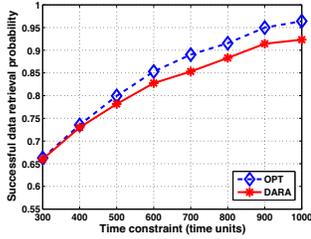


Fig. 3. Comparison of DARA and the optimal solution

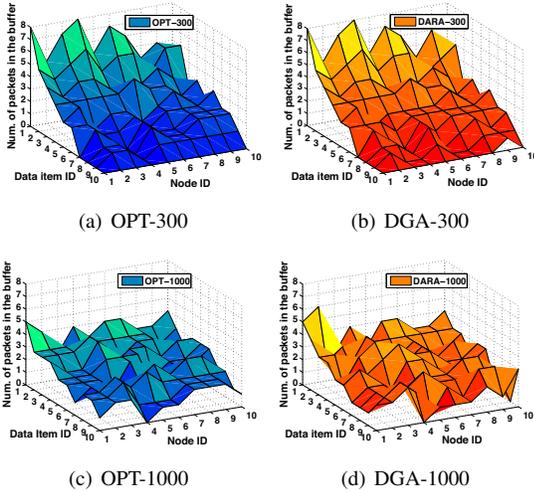


Fig. 4. The data replication result of the optimal solution and DARA

2) *Comparisons to the Optimal Solution:* First, we compare the performance of DARA to the optimal solution derived from the MIP formulation. Due to the extremely high computational complexity of solving MIP, we compare the two schemes using the small-scale trace. Fig. 3 compares the data retrieval probability of the two schemes when the time constraint varies from 300 to 1000 time units. As can be seen from the figure, DARA achieves close performance to the optimal solution without relying on the global knowledge and without suffering from the high computational complexity.

Fig. 4 shows how data is replicated in the two schemes under time constraint of 300 and 1000. When the time constraint is 300 as shown in Fig. 4(a) 4(b), more buffers are allocated to the data items with high popularity (the data item with smaller ID has higher popularity), and the replication results tend to be more skew among the data items. This phenomenon is due to the fact that with short time constraint, a larger replication factor is required for each data to ensure the performance of data access, and the popular data items are prioritized to occupy more buffers. Here the replication factor of a data item denotes the ratio of the total number of packets of the data replicated in the network to the number of packets needed to reconstruct the data. When the time constraint increases to 1000 as shown in Fig. 4(c) 4(d), the storage space is allocated more evenly among the data items. We can also see that the data replication solution derived by DARA is very close to the optimal solution, especially when the time constraint is small.

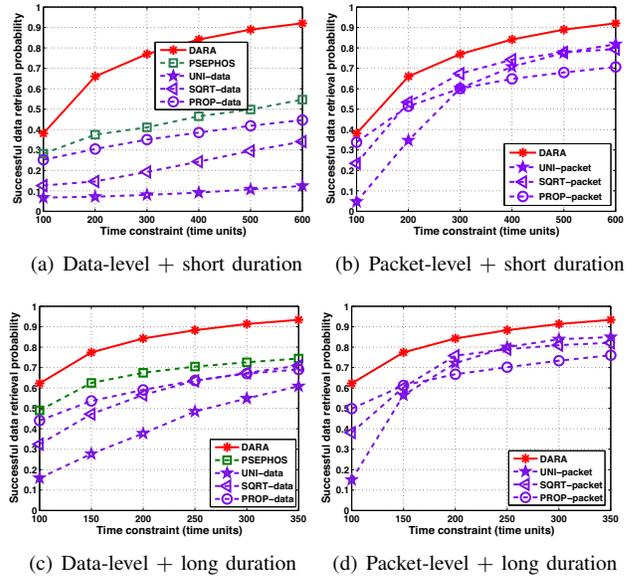


Fig. 5. Comparison of DARA and other schemes on the synthetic traces

3) *Comparisons to Other Replication Schemes:* In this subsection, we use the large-scale trace to evaluate the performance of DARA against seven other schemes, which are divided into two groups: data-level (including UNI-data, SQRD-data, PROP-data, and PSEPHOS) and packet-level (including UNI-packet, SQRD-packet, and PROP-packet). As shown in Fig. 5(a) 5(b), in the scenario of short contact duration, the successful data retrieval probability in the packet-level schemes is much higher than that in the data-level schemes, especially when the time constraint becomes loose. The reason is that packet-level replication schemes can better exploit the contact opportunities. The coded packets of a data item are more evenly allocated among the nodes, which increases the number of potential sources from which a requester can obtain packets. When the time constraint becomes looser, the data requester has more contact opportunities with the nodes replicating the packets. On the contrary, in data-level replication schemes, data packets are replicated in a few hot-spot nodes. The probability to contact those nodes is small, and even the

contact happens, it is hard for those nodes to send out all the packets to the requester during a short contact. Thus, lots of contact opportunities are wasted which significantly reduces the successful data retrieval probability.

As shown in Fig. 5(a), compared to PSEPHOS, UNI-data, SQRT-data, and PROP-data, DARA improves the successful data retrieval probability by 35.1%, 468.9%, 202.5% and 51.2% respectively when the time constraint is 100. Such improvement changes to 68.4%, 639.3%, 169.6% and 106.1% when the time constraint reaches 600. Among the data-level replication schemes, PSEPHOS performs the best, since it considers the node contact patterns, while the other three naive data-level schemes treat all nodes equally.

From Fig. 5(a), it is clear to see that PROP-data performs the best and UNI-data performs the worst among the three naive data-level schemes. However, as shown in Fig. 5(b), among the three naive packet-level schemes, no scheme is obviously better than others. When the time constraint is smaller than 200, PROP-packet performs the best. This is because short time constraint requires for larger replication factor and PROP-packet can at least ensure higher retrieval probability for popular data. When the time constraint is larger than 500, even lower replication factor can ensure good performance, and thus UNI-packet achieves the best performance among the three, where every data item can be allocated with almost enough storage. However, PROP-packet gives too much weight to popular data, which wastes the storage space. Among the three, SQRT-packet achieves the best performance when the time constraint is within the range of [200, 500], since its replication strategy is a trade-off between UNI-packet and PROP-packet. These three packet-level schemes allocate storage only according to the data popularity. Since DARA considers both node contact pattern and data availability, it always has the best performance. Compared to UNI-packet, SQRT-packet, PROP-packet, DARA improves the successful data retrieval probability by 710.4%, 62.4%, and 12.7% with time constraint 100. When the time constraint reaches 600, the improvement changes to 12.5%, 16.0%, and 30.2%.

In the scenario of long contact duration, the performance of packet-level replication schemes has no significant improvement compared to data-level replication schemes as shown in Fig. 5(c) 5(d). When the time constraint is short (within the range of [100, 150]), some data-level schemes such as PSEPHOS even have better performance than naive packet-level schemes. Within short time constraint, the contact opportunities between the data requesters and others are rare, and in the scenario of long contact duration, the expected number of packets can be transmitted during a contact tends to be large. The naive packet-level schemes let the nodes only replicate parts of the data items, which may waste the precious contact opportunities if the contact durations happen to be long. When the time constraint becomes loose (within the range of [200, 350]), the naive packet-level schemes have better performance. Since the data requesters have more contact opportunities with others, the packet-level schemes which

replicate data more evenly among the nodes can make better use of these opportunities. Compared to the seven schemes, DARA still achieves the best performance in the scenario of long contact duration.

C. Realistic Traces

1) *Simulation Settings*: We further evaluate the performance of DARA using two realistic traces: *MIT Reality* [29] and *Infocom05* [30]. The MIT Reality trace was collected by using 97 Nokia 6600 smartphones which were carried by staffs and students at MIT over nine months, and the Infocom05 trace was collected by using 41 iMotes which were carried by the participants of conference Infocom05 over three days. The mobile devices sense each other using Bluetooth and update the contact log at an interval of 300 seconds for MIT Reality trace and 120 seconds for Infocom05 trace. Due to the coarse granularity of the traces, there are many contact records whose contact duration is zero. In the evaluation, we set the sensing interval to be the time unit of transmitting one packet. If the contact duration is zero, we assume that only one packet can be transmitted during the contact.

In the MIT Reality trace (Infocom05 trace, resp.), we set 30 (40, resp.) data items in the network, and each data item contains 64 (32, resp.) packets. Every node has the storage buffer of 128 (96, resp.). We also add a virtual static node to act as the AP. The contact frequency between the AP and mobile nodes is randomly generated within the range of $[2 \times 10^{-7}, 6 \times 10^{-7}]$ and $[8 \times 10^{-6}, 2 \times 10^{-5}]$ in the MIT Reality and the Infocom05 trace, respectively. In each simulation run, the first 1/3 of the trace is used for warmup. The presented results are averaged over 10 runs.

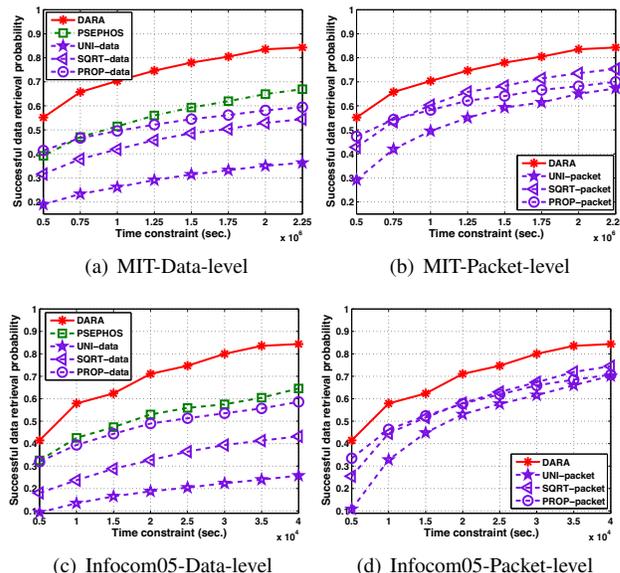


Fig. 6. Comparison of DARA and other schemes on the realistic traces

2) *Results*: Fig. 6 compares the data retrieval probability of DARA to seven other schemes on both the MIT Reality and the Infocom05 trace. As can be seen, packet-level schemes outperform data-level schemes on both traces. Among the three replication strategies: uniform, square-root and proportional,

the uniform strategy performs the worst in both packet-level and data-level schemes, on both traces. The low contact frequency of the trace demands for relatively large replication factor. Since the uniform replication strategy sets equal priority to different data items, most data items are not allocated with enough buffers to ensure high data retrieval probability.

Among data-level replication schemes, PSEPHOS performs the best since it considers different contact patterns of the nodes. Compared to PSEPHOS, DARA improves the successful data retrieval probability by about 33.3% and 33.8% on the MIT Reality and the Infocom05 trace, respectively. Among the naive packet-level replication schemes, SQRT-packet outperforms others in a wide range of time constraint. Compared to it, DARA also improves the successful data retrieval probability by about 15.4% and 25.3% on the MIT Reality and the Infocom05 trace, respectively.

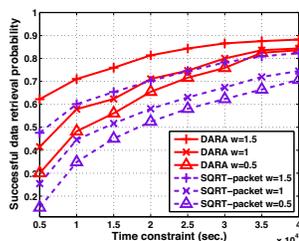


Fig. 7. The effects of the query distribution

Next, we evaluate the impact of the query distribution on DARA by varying parameter w of Zipf distribution. The larger w is, the skewer the query distribution is. When w equals to 0, all data items have uniform popularity. The Infocom05 trace is used. Since SQRT-packet performs almost the best among the seven schemes (see Fig. 6(c) 6(d)), we compare DARA with it when w is set as 0.5, 1, and 1.5. As shown in Fig. 7, skewer query distribution results in better performance. Since the total storage space is not enough to fulfill the demand for all data items, some data items cannot obtain enough storage space. These data items have smaller query rate when the query distribution is skewer, which results in higher data retrieval probability. When w changes from 0.5 to 1.5, DARA always outperforms SQRT-packet.

VII. CONCLUSIONS

In this paper, we studied the impacts of contact duration limitation on data replication in DTNs. Different from traditional data-level replication schemes, we replicate data at packet-level and address the problems of which data items to replicate and how many packets of each data item to replicate at each node. We formulated the packet-level contact duration aware data replication problem as a MIP problem and solved it by CPLEX. Then, we designed a distributed scheme called DARA to replicate data according to a novel “replication benefit” metric, which is determined by the popularity and availability of the data item, and the contribution gain that the node can provide to others. Extensive simulations based on synthetic and realistic traces show that our solution outperforms other replication schemes, and achieves close-to-optimal performance.

REFERENCES

- [1] “Wireless data: The end of all-you-can-eat?” [Online]. Available: http://www.businessweek.com/magazine/content/10_28/b4186034470110.htm
- [2] X. Zhuo, W. Gao, G. Cao, and Y. Dai, “Win-Coupon: An Incentive Framework for 3G Traffic Offloading,” in *Proc. of IEEE ICNP*, 2011.
- [3] B. Han, P. Hui, V. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan, “Cellular traffic offloading through opportunistic communications: A case study,” in *Proc. of ACM CHANTS*, 2010.
- [4] K. Fall, “A Delay-Tolerant Network Architecture for Challenged Internets,” in *Proc. of ACM SIGCOMM*, 2003.
- [5] B. Tang, H. Gupta, and S. Das, “Benefit-based Data Caching in Ad Hoc Networks,” *IEEE TMC*, vol. 7, no. 3, pp. 289–304, 2008.
- [6] I. D. Baev and R. Rajaraman, “Approximation algorithms for data placement in arbitrary networks,” in *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.
- [7] E. Cohen and S. Shenker, “Replication strategies in unstructured peer-to-peer networks,” in *Proc. of ACM SIGCOMM*, 2002.
- [8] J. Zhao and G. Cao, “VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks,” in *Proc. of IEEE INFOCOM*, 2006.
- [9] W. Gao, Q. Li, B. Zhao, and G. Cao, “Multicasting in delay tolerant networks: a social network perspective,” in *Proc. of ACM MOBIHOC*, 2009.
- [10] E. M. Daly and M. Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs,” in *Proc. of ACM MOBIHOC*, 2007.
- [11] J. Reich and A. Chaintreau, “The age of impatience: Optimal replication schemes for opportunistic networks,” in *Proc. of IEEE CONEXT*, 2009.
- [12] S. Ioannidis, L. Massoulie, and A. Chaintreau, “Distributed caching over heterogeneous mobile networks,” in *Proc. of ACM SIGMETRICS*, 2010.
- [13] J. W. Byers, M. Lubys, and M. Mitzenmacher, “A Digital Fountain Approach to Asynchronous Reliable Multicast,” *IEEE JSAC*, vol. 20, no. 8, pp. 1528–1540, 2002.
- [14] “CPLEX: Linear Programming Solver.” [Online]. Available: <http://www.ilog.com/>
- [15] G. Karlsson, V. Ienders, and M. May, “Delay-Tolerant Broadcasting,” in *Proc. of ACM CHANTS*, 2006.
- [16] C. Boldrini, M. Conti, and A. Passarella, “Modelling Data Dissemination in Opportunistic Networks,” in *Proc. of ACM CHANTS*, 2008.
- [17] F. Li and J. Wu, “Mops: Providing Content-based Service in Disruption Tolerant Networks,” in *Proc. of IEEE ICDCS*, 2009.
- [18] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, “Supporting Cooperative Caching in Disruption Tolerant Networks,” in *Proc. of IEEE ICDCS*, 2011.
- [19] X. Zhuo, Q. Li, G. Cao, Y. Dai, B. Szymanski, and T. L. Porta, “Social-Based Cooperative Caching in DTNs: A Contact Duration Aware Approach,” in *Proc. of IEEE MASS*, 2011.
- [20] S. Tewari and L. Kleinrock, “Proportional Replication in Peer-to-Peer Networks,” in *Proc. of IEEE INFOCOM*, 2006.
- [21] A. Balasubramanian, B. N. Levine, and A. Venkataramani, “DTN Routing as a Resource Allocation Problem,” in *Proc. of ACM SIGCOMM*, 2007.
- [22] V. Conan, J. Leguay, and T. Friedman, “Characterizing Pairwise Inter-Contact Patterns in Delay Tolerant Networks,” in *Proc. of Autonomic Computing and Communication Systems*, 2007.
- [23] Y. Lin, B. Li, and B. Liang, “Stochastic Analysis of Network Coding in Epidemic Routing,” *IEEE JSAC*, vol. 26, no. 5, pp. 794–808, 2008.
- [24] —, “Efficient Network Coded Data Transmissions in Disruption Tolerant Networks,” in *Proc. of IEEE INFOCOM*, 2008.
- [25] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks,” in *Proc. of ACM MOBIHOC*, 2008.
- [26] W. Wang, V. Srinivasan, and M. Motani, “Adaptive contact probing mechanisms for delay tolerant applications,” in *Proc. of ACM MOBICOM*, 2007.
- [27] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Pocket switched networks: Realworld mobility and its consequences for opportunistic forwarding,” Tech. Rep. UCAM-CL-TR-617, 2005.
- [28] M. Mitzenmacher and E. Upfal, “Probability and Computing: Randomized Algorithms and Probabilistic Analysis,” Cambridge, 2008.
- [29] N. Eagle and A. Pentland, “Reality Mining: Sensing Complex Social Systems,” *Personal and Ubiquitous Computing*, vol. 10, no. 4, 2006.
- [30] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket Switched Networks and Human Mobility in Conference Environments,” in *Proc. of ACM WDTN*, 2005.