Provenance-driven Data Dissemination in Disruption Tolerant Networks

Mudhakar Srivatsa*, Wei Gao† and Arun Iyengar*
*IBM T. J. Watson Research Center, Hawthorne, NY
†Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA
*{msrivats, aruni}@us.ibm.com †weigao@cse.psu.edu,

***Abstract*—Disruption Tolerant Networks (DTNs) are characterized by low node density, unpredictable node mobility and lack of global network information. Most of the current research efforts in DTNs focus on data forwarding, but only limited work has examined the problem of providing effective access of fused information to mobile users. In this paper, we examine a QoI metric - provenance ("to come from") - which enables a user to establish trust in an information product that is obtained by fusing raw data from one or more information sources (with varying degrees of trust). In particular, we examine a semi-ring model for provenance that allows us to quantify and compare "provenance levels" and propose a novel approach to support provenance queries in DTNs using a cooperative in-network caching approach. To address the challenges of opportunistic network connectivity in DTNs, our basic idea is to intentionally cache both data and its provenance at a set of Network Central Locations (NCLs), which can be easily accessed by other nodes in the network. We propose an effective scheme which ensures appropriate NCL selection based on a probabilistic selection metric, and furthermore coordinate multiple caching nodes to optimize the tradeoff between quality (provenance level), data accessibility and caching overhead. By extensive trace-driven simulations, we show that our proposed caching scheme significantly improves the performance of data access, in terms of the provenance level and data access delay, compared to existing schemes.**

I. Introduction

Disruption Tolerant Networks (DTNs) [9] consist of mobile devices which contact each other opportunistically. Due to the low node density and unpredictable node mobility, only intermittent connectivity among mobile nodes exist in DTNs, and the subsequent difficulty of maintaining persistent end-to-end connection makes it necessary to use "carry-and-forward" methods for data transmission. More specifically, node mobility is exploited to let mobile nodes physically carry data as relays, and forward data opportunistically upon contacts with others. The key problem is therefore how to determine the appropriate relay selection strategy.

Although a large variety of data forwarding schemes have been proposed in DTNs [22], [3], [1], [8], [12], there is only limited research effort on providing effective access of fused information to mobile users[1] in such challenging networks, despite the importance of data accessibility in many mobile computing applications. For example, with the popularization of Smartphones, it is desirable that mobile users can find interesting digital content that is derived by fusing information from one or more nearby peers. In Vehicular Ad-hoc Networks (VANETs), the availability of live traffic information about specific road segments will be beneficial for nearby vehicles to avoid traffic delays.

In such applications wherein information sources with diverse trust levels contribute towards a fused information product, we argue that it is important to examine provenance as a QoI (Quality of Information) metric, in addition to classical performance metrics such as storage, bandwidth and latency. In this paper we examine a semi-ring provenance model [14] which encodes provenance metadata as Boolean monotone expressions. In this model, $d = (a \wedge b) \vee c$ indicates that the provenance of an information product d can be established using a and b or independently using c. This semi-ring model leads to a partial order on provenance levels for d: $\{\} < \{a, b\} = \{c\} < \{a, b, c\}$, namely, retrieving only data items a or b has lesser utility than retrieving $\{a, b\}$ or $\{c\}$, which in turn has lesser utility than retrieving all $\{a, b, c\}$ (because we now have two independent corroborating evidences for d).

In this paper we examine the problem of supporting of support provenance queries over DTNs. A common technique used to improve the performance of data access is caching. The basic idea is to cache data at appropriate network locations based on the query history, so that queries in the future can be responded with less delay. Although cooperative caching has been extensively studied for both web-based applications [10], [26] and wireless ad-hoc networks [27] to allow the sharing and coordination of cached data among multiple nodes, it is difficult to be realized in DTNs due to the lack of persistent network connectivity. Further, in the context of provenance queries (unlike data queries), the answer to a query is not a unique data item; indeed, as shown in the example above, it may be possible to return different sets of data items with the goal of achieving different provenance levels.

In this paper, we propose a novel scheme to address the aforementioned challenges and to effectively support provenance queries over DTNs. Our basic idea is to intentionally cache data and its provenance at a set of Network Central Locations (NCLs), each of which corresponds to a group of mobile nodes being easily accessed by other nodes in the network. More specifically, each NCL is represented by a central node, which has high popularity in the network and is prioritized for caching data. Due to the limited caching buffer of central nodes, multiple nodes near a central node may be involved in caching, and we ensure that popular data (and its provenance) is always cached nearer to the central nodes via dynamic cache replacement based on the query history.

The rest of this paper is organized as follows. In Section II we briefly review existing work. Section III provides an

[1]In the rest of this paper, "node" and "user" are used interchangeably.

overview of our approach and highlights our motivation of intentional caching in DTNs. Section IV describes how to appropriately select NCLs in DTNs, and Section V describes the details of our proposed caching scheme. The results of trace-driven performance evaluations are shown in Section VI, and Section VII concludes the paper.

II. Related Work

Research on data forwarding in DTNs originates from Epidemic routing [25] which floods the entire network. Some later studies develop relay selection strategies to approach the performance of Epidemic routing with lower forwarding cost, based on the prediction of node contact in the future. Some schemes do such prediction by estimating node co-location probabilities based on their mobility patterns, which are characterized by Kalman filter [6] or semi-Markov chains [28]. Some others [3], [1] exploit node contact records in the past as stochastic process for better prediction accuracy, based on the experimental [5], [18] and theoretical [4] analysis on node contact characteristics. The social network properties of human mobility, such as the centrality and community structures, are also exploited for data forwarding decision in recent social-based forwarding schemes [7], [16], [12].

Caching is popular approach to provide data accessibility to mobile users in DTNs. [27] studied cooperative caching in wireless ad-hoc networks, in which each node caches pass-by data based on the data popularity, so that queries in the future can be responded by the caching node with less delay. In other words, the caching locations are selected incidentally among all the nodes in the network. Some research efforts [20], [15] have also been made for caching in DTNs, but the proposed caching strategies only improve data accessibility from the infrastructure network, such as WiFi Access Points (APs) [15] or Internet gateways [20]. Recent work has explored peer-to-peer data sharing and access among mobile users themselves, but only simple (provenance agnostic) data queries are only considered [11].

Several models of provenance have been proposed by past work [13], [23], [24]. However, most of the past works on provenance models were in the context of traditional databases. Consequently, they do not adequately account for networking constraints (e.g., disconnected environment) that arise in DTNs. This paper, to the best of our knowledge, presents the first attempt towards examining provenance queries over DTNs with the goal of quantifying tradeoffs between provenance levels (using a semi-ring model for provenance [14]) and access delays.

III. Overview

A. Network Modeling

Opportunistic node contacts in DTNs are described by the network *contact graph* $G(V, E)$, where the stochastic contact process between a node pair $i, j \in V$ is modeled as an edge e_{ij}. We assume that node contacts are symmetric; i.e., node j contacts i whenever i contacts j, and the network contact graph is therefore undirected.

The characteristics of an edge $e_{ij} \in E$ are mainly determined by the properties of inter-contact time among mobile nodes. Similar to previous work [1], [19], [17], [29], we consider the pairwise node inter-contact time as exponentially distributed. The contacts between nodes i and j then form a Poisson process with the contact rate λ_{ij}, which remains relatively constant and is calculated at real-time from the cumulative contacts between nodes i and j in a time-average manner.

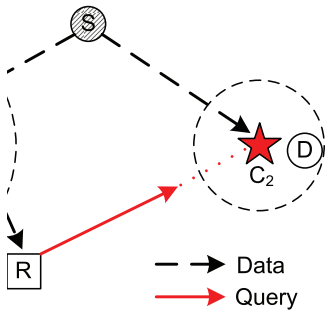


Fig. 1. Intentional Caching

B. In-Network Caching

In this paper, we consider a general application scenario for cooperative caching in DTNs, in which each node may generate data with a globally unique identifier and specific lifetime, and may also request for another data by sending queries with a finite time constraint. Therefore, data requesters are randomly distributed in the network, and are not spatially correlated with each other. The network is responsible for delivering the requested data with its provenance; in doing so, the network may trade off storage and data access delay overhead with provenance level. We assume that each node has only limited space for caching, and our objective is to effectively utilize the available space to optimize the overall caching performance.

Our basic idea is to intentionally cache data only at a specific set of NCLs, which can be easily accessed by other nodes in the network. Correspondingly, queries are forwarded to these NCLs for data access. The big picture of our proposed caching scheme is illustrated in Figure 1. Each NCL is represented by a central node[2], which corresponds to a star in Figure 1. The push and pull caching strategies conjoin at the NCLs. The data source S actively pushes its generated data towards the NCLs; if the buffer of a central node C_1 is full, data is cached at one or more nodes near a NCL (e.g., A near C_1). Correspondingly, the requester R pulls the data by querying the NCLs, and data copies from multiple NCLs are returned to the requester, in order to ensure data accessibility within the time constraint of the query.

IV. Network Central Locations

In this section, we describe how to appropriately select NCLs based on a probabilistic metric evaluating the data

[2]In the rest of this paper, a central node is used equivalently to denote the corresponding NCL.

TABLE I
TRACE SUMMARY

Trace	Infocom05	Infocom06	MIT Reality	UCSD
Network type	Bluetooth	Bluetooth	Bluetooth	WiFi
# devices	41	78	97	275
# internal contacts	22,459	182,951	114,046	123,225
Duration (days)	3	4	246	77
Granularity (secs)	120	120	300	20
Pairwise contact frequency (per day)	4.6	6.7	0.024	0.036

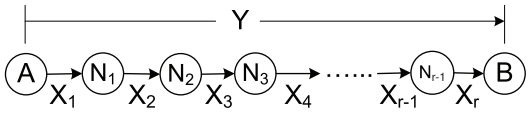


Fig. 2. Opportunistic path

transmission delay among mobile nodes in DTNs. The applicability of such selection in practice is then validated by the heterogeneity of node contact pattern in realistic DTN traces.

A. NCL Selection Metric

In order to develop an appropriate metric for NCL selection, we first define the multi-hop opportunistic connection on the network contact graph $G = (V, E)$.

Definition 1: Opportunistic path

A r-hop opportunistic path $P_{AB} = (V_P, E_P)$ between nodes A and B consists of a node set $V_P = \{A, N_1, N_2, ..., N_{r-1}, B\} \subset V$ and an edge set $E_P = \{e_1, e_2, ..., e_r\} \subset E$ with edge weights $\{\lambda_1, \lambda_2, .., \lambda_r\}$. The path weight is the probability $p_{AB}(T)$ that a data item is opportunistically transmitted from A to B along P_{AB} within time T.

An opportunistic path is illustrated in Figure 2. The inter-contact time X_k between nodes N_k and N_{k+1}, as a random variable, follows an exponential distribution with probability density function (PDF) $p_{X_k}(x) = \lambda_k e^{-\lambda_k x}$. Hence, the total time needed to transmit data from A to B is $Y = \sum_{k=1}^{r} X_k$ following a hypoexponential distribution [21], such that

$$p_Y(x) = \sum_{k=1}^{r} C_k^{(r)} p_{X_k}(x), \tag{1}$$

where the coefficients $C_k^{(r)} = \prod_{s=1, s\neq k}^{r} \frac{\lambda_s}{\lambda_s - \lambda_k}$.

From Eq. (1), the path weight is written as

$$p_{AB}(T) = \int_0^T p_Y(x)dx = \sum_{k=1}^{r} C_k^{(r)} \cdot (1 - e^{-\lambda_k T}), \tag{2}$$

and the data transmission delay between two nodes A and B is measured by the weight of the shortest opportunistic path between the two nodes.

The metric C_i for a node i to be selected as a central node to represent a NCL is then defined as follows:

$$C_i = \frac{1}{N-1} \cdot \sum_{j=1, j\neq i}^{N} p_{ij}(T), \tag{3}$$

where N is the total number of nodes in the network. This metric indicates the average probability that data can be transmitted from a random node in the network to node i within time T, and therefore can also be considered as indicating the average distance from a random node in the network to node i.

In practice, the top K nodes with the highest metric values are selected by the network administrator as the central nodes of NCLs, and such NCL selection is done before any data access operation. A network warm-up period is needed for the administrator to collect information about the pairwise node contact rate, and to calculate the weight of opportunistic paths among mobile nodes. The NCL information is notified by the administrator to each node in the network, and a node maintains its shortest opportunistic path to each NCL. We assume that the set of NCL nodes remain stable over time; this stability is also validated in various mobility scenarios [12], [29]. As a result, the selected NCLs will not be changed during data access.

B. Trace-based Validation

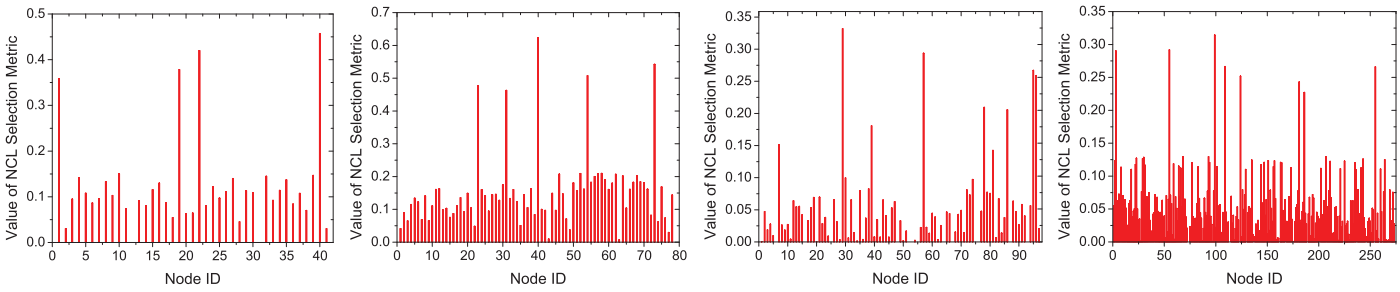
The practical applicability of NCL selection is based on the heterogeneity of node contact patterns. In this section, we validate this applicability using realistic DTN traces. These traces record contacts among users carrying hand-held mobile devices at a technical conference and a university campus. The devices equipped with a Bluetooth interface periodically detect their peers nearby, and a contact is recorded when two devices move close to each other. The devices equipped with a WiFi interface search for nearby WiFi Access Points (APs) and associate themselves to the APs with the best signal strength. A contact is recorded when two devices are associated to the same AP. The traces are summarized in Table I.

In order to calculate the weight of an opportunistic path according to Eq. (2), we calculate the pairwise contact rates based on the cumulative contacts between each pair of nodes during the entire trace. According to Eq. (2), inappropriate values of T will make C_i close to 0 or 1. Therefore, due to the heterogeneity of the pairwise contact frequency in different traces, different values of T are used adaptively chosen; T is set as 1 hour for the two Infocom traces, 1 week for the MIT Reality trace, and 3 days for the UCSD trace.

The results in Figure 3 show that the distributions of NCL selection metric values of mobile nodes are highly skewed in all traces, such that the metric values of a few nodes are much higher than that of other nodes. This difference can be up to tenfold in some traces, and validates that our proposed NCL selection metric appropriately reflects the heterogeneity of node contact patterns. As a result, the selected NCLs can be easily accessed by other nodes in the network, which hence ensures the performance of our proposed caching scheme.

V. PROVENANCE CACHING

In this section, we present our provenance caching scheme in detail. Our basic idea is to intentionally cache data and provenance at a set of NCLs, which can be promptly accessed by other nodes in the network. The functionality of our scheme consists of the following three components:



(a) Infocom05 (b) Infocom06 (c) MIT Reality (d) UCSD

Fig. 3. Values of NCL selection metric on realistic DTN traces

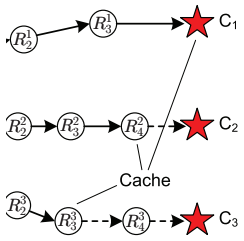


Fig. 4. Determining caching location at NCLs

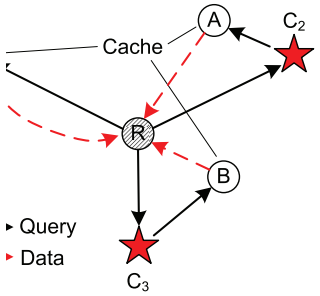


Fig. 5. Pulling data from the NCLs

1) When a data source generates new data, it pushes the data and its provenance expressions (in the semi-ring model) to the central nodes of NCLs which are prioritized to cache data. The NCL node stores a copy of all provenance expressions. One copy of data is cached at each NCL; if the caching buffer of a central node is full, another node near the central node will be decided to cache the data. Such decisions are automatically made based on the buffer conditions of nodes involved in the pushing process.

2) A requester multicasts a query to the central nodes of NCLs to pull the data, and a central node forwards the query for the data (along with its provenance expression) to the nodes caching the required data items. A number of cached data copies are returned to the requester, and we optimize the tradeoff between provenance level, data accessibility and transmission overhead by probabilistically controlling the number of returned data copies.

3) Utility-based[3] cache replacement is conducted whenever two caching nodes contact each other, and ensures that high utility data is cached nearer to the central nodes. We present an approach to quantify utility of a data item based on its marginal impact on the provenance of the requested data and the popularity of the requested data itself.

Figure 4 and 5 illustrates the first two steps described above. In the rest of this section, due to lack of space, we examine the utility-based caching scheme in detail.

A. Utility-based Caching

There are two major components that contribute towards the utility of a data item: data popularity and marginal provenance level.

*1) **Data Popularity**:* The popularity of a data item in the network is probabilistically estimated based on the occurrences of the past k requests to this data, which happened during the time period $[t_1, t_k]$. We assume that such occurrences of data requests in the past follow a Poisson distribution with the parameter $\lambda_d = k/(t_k - t_1)$, and data popularity is defined as the probability that this data will be requested again in the future before the data expires. If data d_i expires at time t_e, the popularity w_i of d_i is written as

$$w_i = 1 - e^{-\lambda_d \cdot (t_e - t_k)}, \tag{4}$$

Which is actually the probability that d_i is requested at least once again in the future before time t_e. To calculate the popularity of a data item, a node needs to recursively maintain two time values about the past occurrences of data requests, and therefore will only incur negligible space overhead.

*2) **Marginal Provenance Level**:* Besides the popularity of a data item, we quantify the vitality of a data item in establishing the provenance of the requested data item. In order to do so, we examine the provenance expression of data items. For example, given a provenance expression $d = (a \wedge b) \vee c$, the data item c is more important in establishing the provenance of d than data items a or b. In general, we denote a provenance expression as a boolean monotone function $d = f(a_1, \cdots, a_n)$. The contribution of data item a_i towards the provenance of d is quantified as:

$$w_i^d = \frac{\#f(a_1, \cdots, a_{i-1}, 1, a_{i+1}, \cdots, a_n)}{2^{n-1}} \tag{5}$$

[3]For newly created data, the utility value will initially be low since the data has not yet been requested.

Where the notation a_i is overloaded to denote both the data item and 0/1 Boolean variable and $\#f$ denotes the number of satisfiable assignments to boolean variables $\{a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n\}$ such that f evaluates to true. For example, given $d = (a \wedge b) \vee c$ the marginal provenance levels are given as $w_a^d = w_b^d = \frac{1}{2}$ and $w_c^d = 1$; given $d = a \wedge (b \vee c)$ the marginal provenance levels are given as $w_b^d = w_c^d = \frac{1}{2}$ and $w_a^d = \frac{3}{4}$. We also set $w_*^d = 0$ for all data items that do not contribute to the provenance of d.

We note that while counting the number of satisfiable assignments to a general boolean expression is a NP-hard problem; however, counting such assignments over monotone boolean expressions is relatively easier. The key observation here is that if for some assignment of $\{a_i\}$, $f(a_1, \cdots, a_n)$ evaluates to true, then for all $a_i = 0$ in the satisfiable assignment setting $a_i = 1$ retains the satisfiability of f (due to the monotone property).

*3) **Overall Utility**:* The overall utility of a data item d depends upon its popularity and the popularity of all data items d' such that the marginal provenance level $w_d^{d'} > 0$. Further, assuming finite cache space we set the overall utility as being inversely proportional to the size of the data item d. Hence, the overall popularity of d and its utility is given by:

$$pop(d) = w_d + \sum_{d' \neq d} w_{d'} * w_d^{d'}$$

$$u_d = \frac{pop(d)}{size(d)}$$

In our experimental section, we compare our approach with that of Bundle Cache [20] and Cache Data [27] that are agnostic to provenance. However, these approaches account for the popularity of the data item (as shown in Equation 6), albeit the marginal provenance level; we set $w_d^{d'} = 1$ if the provenance of d' depends upon d (and 0 otherwise) without explicitly taking the provenance expression. For example, given a provenance expression $d = a \wedge (b \vee c)$, in provenance agnostic caching approach we set $w_a^d = w_b^d = w_c^d = 1$ and w_x^d = (for all $x \notin \{a, b, c\}$).

*4) **Caching Policy**:* Given an opportunistic contact between two nodes in the DTN, the nodes collectively examine all the data items available in their contacts. Each node caches a data item d (from the aggregate pool of data items) with probability that is proportional to its utility u_d; this process is repeated until the cache node runs out of storage space. We observe that high utility items are likely to be cached on both nodes; while low utility items may fade out from the caches.

B. Answering Provenance Queries

As shown in Figure 5, due to the probabilistic nature of data delivery in DTNs, multiple data copies are replied to the requester from NCLs to ensure that the requester is able to receive data before the query expires. However, only the first data copy received by the requester is useful, and all the others are essentially useless and waste the network bandwidth. This problem is further complicated since there may be multiple ways to establish the provenance of the requested data item – there is evidently a tradeoff between provenance level and networking overhead. In this section, we propose a probabilistic scheme to address these challenges and optimize the tradeoff between provenance level and transmission overhead. Our basic idea is that, having received the query, a caching node probabilistically decides whether to return the cached data to the requester.

We assume that the query is generated with a time constraint T_q, and it takes $t_0 < T_q$ for the query to be forwarded from requester R to caching node C. If there is no tight constraint on the network storage and bandwidth, each node is able to maintain the information about the shortest opportunistic paths to all the other nodes in the network. According to Eq. (2 and 5), C can determine whether to reply data to R with the probability $w_d^q * p_{CR}(T_q - t_0)$, where w_d^q denotes the marginal provenance level of d with respect to query q and $p_{CR}(T_q - t_0)$ denotes the probability that the data can be transmitted from C to R within the remaining time $T_q - t_0$.

Otherwise, a node only maintains the information about the shortest opportunistic paths to the central nodes, and it is difficult for C to estimate the data transmission delay to R. Instead, the probability for deciding the data response is calculated only based on the remaining time $T_q - t_0$ for responding to the query and the marginal provenance level of a data item with respect to the queried data item. In general, this probability should be inversely proportional to $T_q - t_0$, and we calculate this probability as a Sigmoid function $p_R(t)$, where $p_R(T_q) = p_{\max} \in (0, 1]$ and $p_R(0) = p_{\min} \in (p_{\max}/2, p_{\max})$. This function is written as

$$p_R(t) = \frac{k_1 * w_d^q}{1 + e^{-k_2 \cdot t}}, \tag{6}$$

where $k_1 = 2p_{\min}$, $k_2 = \frac{1}{T_q} \cdot \ln(\frac{p_{\max}}{2p_{\min} - p_{\max}})$. Our approach allows the user to tune the provenance level of a query response by suitably selecting parameters $p_{\max}$ and $p_{\min}$ (the maximum and minimum response probabilities).

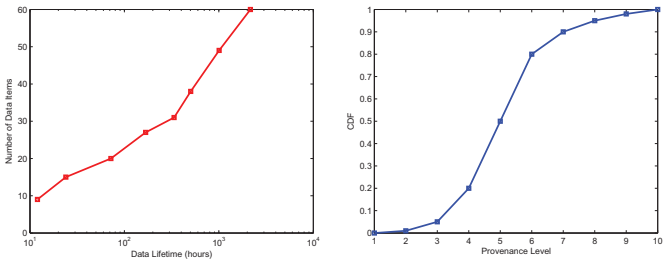
VI. Performance Evaluation

In this section, we evaluate the performance of our proposed provenance-driven caching scheme, which is compared with the following data access schemes:

- **No Cache**, in which caching is not used for data access, and each query result is returned only by the data source.
- **Random Cache**, in which every requester caches the received data to facilitate data access in the future.
- **CacheData** [27], which is proposed for cooperative caching in wireless ad-hoc networks, and lets each selected relay in DTNs cache the pass-by data according to their popularity.
- **Bundle Cache** [20], which packs network data as bundles, and makes caching decision on pass-by data by considering the node contact pattern in DTNs, so as to minimize the average data access delay.

The following metrics are used for evaluations. Each simulation is repeated multiple times with randomly generated data and queries for statistical convergence.

- **Provenance Level** of a query response, defined as the sum of the marginal provenance levels of all the data



(a) Amount of network data (b) Max Provenance Level CDF

Fig. 6. Experiment setup

items retrieved. Recall that given $d = a \wedge (b \vee c)$ the marginal provenance levels are given as $w_b^d = w_c^d = \frac{1}{2}$ and $w_a^d = \frac{3}{4}$. Hence, given a query for the provenance of d, if the network responds with data items a and b then the provenance level of such a response is $w_a^d + w_b^d = \frac{5}{4}$.

- **Data access delay**, the average delay for getting responses to queries.
- **Caching overhead**, the average number of data copies being cached in the network.
- **Access Bandwidth**, the average number bytes required to answer provenance queries.
- **Number of NCLs**, the number of network central locations.

A. Experiment Setup

Our performance evaluations are performed on the *Infocom06* and *MIT Reality* traces collected from realistic DTNs, and the details of the two traces are summarized in Table I. We assume that each node is able to communicate with other nodes in contact through bidirectional wireless links with a capacity of 2.1Mb/s (Bluetooth EDR). In all the experiments, a node updates its contact rates with other nodes in real time based on the up-to-date contact counts since the network starts, and furthermore maintains the information about the shortest opportunistic path to the central nodes. The first half of the trace is used as the warm-up period for the accumulation of network information and subsequent NCL selection, and all the data and queries are generated during the second half of the trace.

*1) **Data Generation**:* Each node in the network periodically checks whether it has generated data which has not expired yet. If not, the node determines whether to generate new data with a unified probability p_G. Each generated data has finite lifetime whose value is uniformly distributed in the range $[0.5T_L, 1.5T_L]$, and the period for data generation decision is also set as the average data lifetime T_L. For simplicity, in our evaluations we fix $p_G = 0.2$, and the amount of data in the network is hence controlled by T_L, as illustrated in Figure 6(a) for the *MIT Reality* trace.

Provenance expressions from a medical dataset [13] are assigned to the data items generated above. Table II shows sample provenance expressions from our dataset, with possible query responses and its provenance level; note that maximum provenance level (shown within brackets) corresponds to a query response that retrieves all the data items in the provenance expression. Figure 6(b) shows the CDF of provenance levels of data items from our dataset.

The data size is uniformly distributed in the range $[0.5s_{avg}, 1.5s_{avg}]$, and the caching buffer of nodes is uniformly distributed in the range $[200\text{Mb}, 600\text{Mb}]$. In the experiments, the parameter s_{avg} is adjusted to simulate different node buffer conditions.

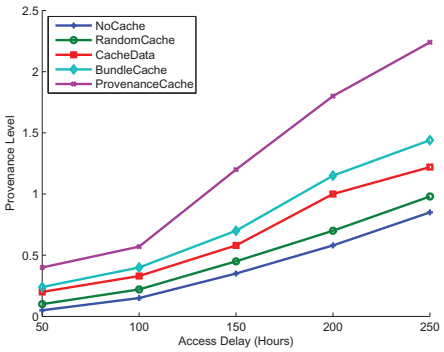
*2) **Query Pattern**:* Queries are randomly generated at all the nodes in the network, and each query is associated with a delay constraint. We assume that the query pattern follows a Zipf distribution, which has been proved to appropriately describe the query pattern of web data access [2]. More specifically, Let $P_j \in [0, 1]$ be the probability that data j is requested, and M be the total number of data items in the network, we have $P_j = \frac{1/j^s}{\sum_{i=1}^{M} 1/i^s}$, where s is an exponent parameter ($s = 1$ in our experiments).

B. Experiment Results

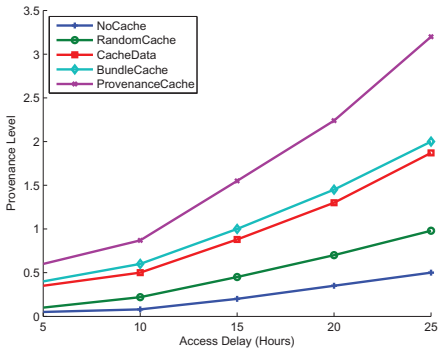
We first evaluate the caching performance of our scheme using the *MIT Reality* and the *Infocom06* trace. We set the number (K) of NCLs as 8 (and 5) according to the trace-based validation results shown in Figure 3(c) (and Figure 3(b) resp.), and generate the query pattern following the Zipf distribution with exponent $s = 1$. By default, the average data lifetime T_L is set as 1 week, and the average data size s_{avg} is set as 100Mb. These parameters are then adjusted for different performance evaluation purposes. Figure 7 shows the tradeoff between provenance level (of a query response) and the access delay. In this experiment we restrict the average number of data item replicas to three (note that this is the average: hence, some high utility data items may have a much larger number of replicas) and the access bandwidth is unlimited (i.e., we do not restrict the number of bytes that may be exchanged between two nodes on an opportunistic contact). We observe that given more time, all approaches generally can improve the quality of their answers. However, Figure 7 shows that provenance caching approach proposed in this paper vastly outperforms the state-of-the-art approaches in terms of the quality (provenance level) of the query responses.

Figure 8 shows the tradeoff between provenance level (of a query response) and cache overhead that is quantified as the average number of replicas of a data item in the network. In this experiment we restrict the access delay to at most 100 hours (and 10 minutes) for the *MIT Reality* dataset (and *Infocom06* dataset resp.). We also assume that the access bandwidth is unlimited. We observe that our utility-based caching approach that accounts for both the data popularity and the marginal provenance level of a data item with respect to a query, performs significantly better that past approaches that are agnostic to provenance. We note that past approaches such as BundleCache and CacheData account for the popularity of the data items, but not the marginal provenance level of a data item.

Figure 9 shows the tradeoff between provenance level and the access bandwidth that is quantified as the average number of bytes exchanged between nodes in order to respond to a

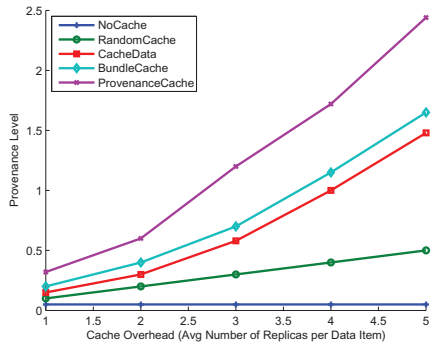


(a) MIT Reality Data Set

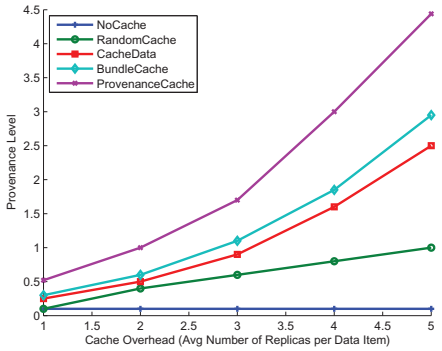


(b) Infocom06 Data Set

Fig. 7. Access Delay

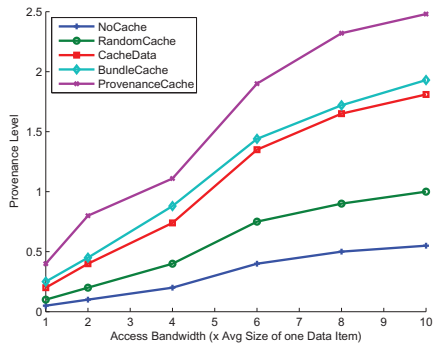


(a) MIT Reality Data Set

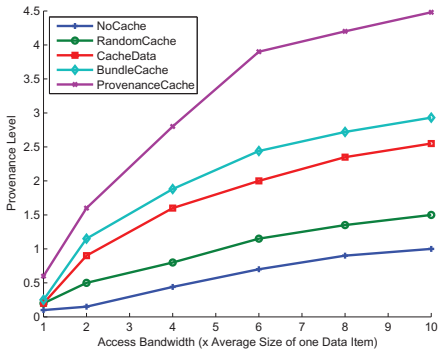


(b) Infocom06 Data Set

Fig. 8. Caching Overhead

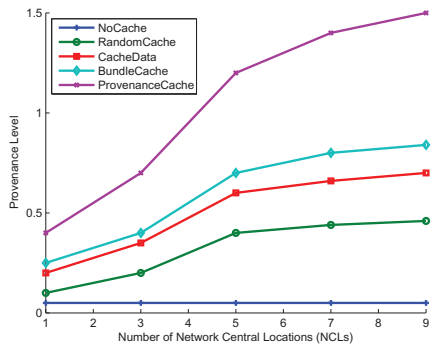


(a) MIT Reality Data Set

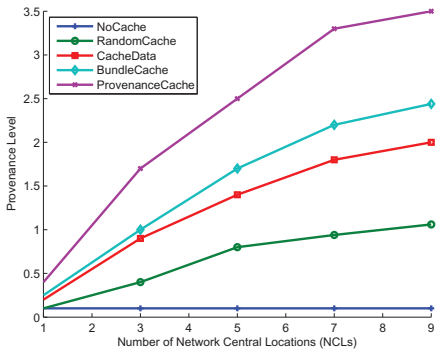


(b) Infocom06 Data Set

Fig. 9. Access Bandwidth



(a) MIT Reality Data Set



(b) Infocom06 Data Set

Fig. 10. # NCLs

TABLE II
SAMPLES FROM EXPERIMENTAL DATASET

Sample Provenance Expressions	Sample Query Response	Provenance Level of Response (Max Provenance Level)
$d_{02b8} \vee (d_{959a} \wedge d_{c3d2})$	$\{d_{02b8}, d_{959a}\}$	1.5 (2.0)
$(d_{02b8} \wedge d_{3fa7}) \vee (d_{959a} \wedge d_{aa4c})$	$\{d_{3fa7}, d_{aa4c}\}$	1.0 (2.0)
$d_{02b8} \vee d_{aa4c}$	$\{d_{02b8}, d_{aa4c}\}$	2.0 (2.0)
$(d_{02b8} \vee d_{aa4c}) \wedge (d_{959a} \vee d_{c3d2}) \wedge (d_{3fa7} \vee d_{aa4c})$	$\{d_{02b8}, d_{c3d2}, d_{3fa7}\}$	1.6875 (3.375)
$(d_{02b8} \wedge d_{959a}) \vee d_{c51a} \vee (d_{8ee5} \wedge d_{4981}) \vee (d_{8ee5} \wedge d_{c3d2}) \vee (d_{3fa7} \wedge d_{aa4c})$	$\{d_{c51a}, d_{8ee5}, d_{3fa7}, d_{4981}\}$	2.5 (5.0)

query. We set the caching over to three and the access delay to 100 hours (and 10 minutes) for the *MIT Reality* dataset (and *Infocom06* dataset resp.). Given more access bandwidth all nodes in the network that hold either the data or its provenance responds to the query − however, we note that only the first receipt of a data item at the querier is useful. Figure 9 shows that the provenance caching approach makes the best use of the available access bandwidth by offering higher provenance level in the query response when compared to other approaches.

Figure 10 shows the tradeoff between provenance level and the number of network central locations. When the number of NCLs is small it generally takes longer to forward queries and data between the requesters and caching nodes, and hence the performance of data access is reduced. In contrast, when the number of NCLs is large, further increase will not improve the performance of data access, because the newly selected central nodes are essentially not good at communicating with other nodes in the network.

VII. CONCLUSIONS

In this paper, we propose a novel scheme to support provenance queries in DTNs, in order to provide effective access of fused information to mobile users. Our basic idea is to intentionally cache data and its provenance at a pre-specified set of NCLs in the network, which can be easily accessed by other nodes. We propose an effective scheme which ensures appropriate NCL selection based on a probabilistic selection metric, and furthermore coordinates multiple caching nodes to optimize the tradeoff between provenance level, data accessibility and caching overhead. Extensive trace-driven simulations show that our scheme significantly improves the provenance level in query responses and reduces data access delay, compared with existing caching schemes.

ACKNOWLEDGEMENT

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proceedings of SIGCOMM*, pages 373–384. ACM New York, NY, USA, 2007.

[2] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of INFOCOM*, volume 1, 1999.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. *Proc. INFOCOM*, 2006.

[4] H. Cai and D. Y. Eun. Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet. *Proc. MobiCom*, pages 159–170, 2007.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Trans. on Mobile Computing*, 6(6):606–620, 2007.

[6] P. Costa, C. Mascolo, M. Musolesi, and G. Picco. Socially Aware Routing for Publish-Subscribe in Delay-Tolerant Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 26(5):748–760, 2008.

[7] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. *Proc. MobiHoc*, 2007.

[8] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot. Delegation Forwarding. *Proc. MobiHoc*, 2008.

[9] K. Fall. A delay-tolerant network architecture for challenged internets. *Proc. SIGCOMM*, pages 27–34, 2003.

[10] L. Fan, P. Cao, J. Almeida, and A. Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.

[11] W. Gao, A. Iyengar, M. Srivatsa, and G. Cao. Supporting cooperative caching in disruption tolerant networks. In *ICDCS*, 2011.

[12] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: a social network perspective. In *Proceedings of MobiHoc*, pages 299–308, 2009.

[13] F. Geerts, A. Kementsietsidis, and D. Milano. Mondrian: Annotating and querying databases through colors and blocks. In *ICDE*, 2006.

[14] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *SIGMOD*, 2007.

[15] Y. Huang, Y. Gao, K. Nahrstedt, and W. He. Optimizing File Retrieval in Delay-Tolerant Content Distribution Community. In *Proceedings of the Int'l Conference on Distributed Computing Systems (ICDCS)*, pages 308–316, 2009.

[16] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. *Proc. MobiHoc*, pages 241–250, 2008.

[17] S. Ioannidis, A. Chaintreau, and L. Massoulie. Optimal and scalable distribution of content updates over a mobile social network. *Proc. INFOCOM*, 2009.

[18] T. Karagiannis, J.-Y. Boudec, and M. Vojnovic. Power law and exponential decay of inter contact times between mobile devices. *Proc. MobiCom*, pages 183–194, 2007.

[19] U. Lee, S.-Y. Oh, K.-W. Lee, and M. Gerla. Scalable multicast routing in delay tolerant networks. *Proc. ICNP*, 2008.

[20] M. J. Pitkanen and J. Ott. Redundancy and distributed caching in mobile dtns. In *Proceedings of 2nd ACM/IEEE Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*. ACM, 2007.

[21] S. M. Ross. *Introduction to probability models*. Academic Press, 2006.

[22] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, 2005.

[23] D. Srivastava and Y. Velegrakis. Intentional associations between data and metadata. In *ACM SIGMOD*, 2007.

[24] W. C. Tan. Provenance in databases: Past, current, and future. In *IEEE Data Engineering Bulletin, 30(4): 3-12*, 2007.

[25] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. *Technical Report CS-200006, Duke University*, 2000.

[26] D. Wessels and K. Claffy. ICP and the Squid Web Cache. *IEEE Journal on Selected Areas in Communications (JSAC)*, 16(3):345–357, 2002.

[27] L. Yin and G. Cao. Supporting Cooperative Caching in Ad Hoc Networks. *IEEE Trans. on Mobile Computing*, 5(1):77–89, 2006.

[28] Q. Yuan, I. Cardei, and J. Wu. Predict and relay: an efficient routing in disruption-tolerant networks. In *Proc. MobiHoc*, pages 95–104, 2009.

[29] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni. Recognizing Exponential Inter-Contact Time in VANETs. In *Proceedings of INFOCOM*, 2010.