**INTRODUCTION**

DISRUPTION tolerant networks (DTNs) consist of mobile devices that contact each other opportunistically. Due to the low node density and unpredictable node mobility, only intermittent network connectivity exists in DTNs, and the subsequent difficulty of maintaining end-to end communication links makes it necessary to use “carry and-forward” methods for data transmission. Examples of such networks include groups of individuals moving in disaster recovery areas, military battlefields, or urban sensing applications. In such networks, node mobility is exploited to let mobile nodes carry data as relays and forward data opportunistically when contacting others. The key problem is, therefore, how to determine the appropriate relay selection strategy.

Although forwarding schemes have been proposed in DTNs, there is limited research on providing efficient data access to mobile users, despite the importance of data accessibility in many mobile applications. For example, it is desirable that smart phones users can find interesting digital content from their nearby peers. In vehicular ad-hoc networks (VANETs), the availability of live traffic information will be beneficial for vehicles to avoid traffic delays. In these applications, data are only requested by mobile users whenever needed, and requesters do not know data locations in advance. The destination of data is, hence, unknown when data are generated. This communication paradigm differs from publish/subscribe systems in which data are forwarded by broker nodes to users according to their data subscriptions. Appropriate network design is needed to ensure that data can be promptly accessed by requesters in such cases.

A common technique used to improve data access performance is caching, i.e., to cache data at appropriate network locations based on query history, so that queries in the future can be responded with less delay. Although cooperative caching has been studied for both web-based applications and wireless ad hoc networks to allow sharing and coordination among multiple caching nodes, it is difficult to be realized in DTNs due to the lack of persistent network connectivity. First, the opportunistic network connectivity complicates the estimation of data transmission delay, and furthermore makes it difficult to determine appropriate caching locations for reducing data access delay. This difficulty is also raised by the incomplete information at individual nodes about query history. Second, due to the uncertainty of data transmission, multiple data copies need to be coached at different locations to ensure data accessibility. The difficulty in coordinating multiple caching nodes makes it hard to optimize the tradeoff between data accessibility and caching overhead.

In this paper, we propose a novel scheme to address the aforementioned challenges and to efficiently support cooperative caching in DTNs. Our basic idea is to intentionally cache data 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. 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 for caching, and we ensure that popular data are always cached nearer to the central nodes via dynamic cache replacement based on query history. Our detailed contributions are listed as follows:

* We develop an efficient approach to NCL selection in DTNs based on a probabilistic selection metric. The selected NCLs achieve high chances for prompt response to user queries with low overhead in network storage and transmission.
* We propose a data access scheme to probabilistically coordinate multiple caching nodes for responding to user queries. We furthermore optimize the tradeoff between data accessibility and caching overhead, to minimize the average number of cached data copies in the network.
* We propose a utility-based cache replacement scheme to dynamically adjust cache locations based on query history, and our scheme achieves good tradeoff between the data accessibility and access delay.