Cooperative File Sharing in Hybrid Delay Tolerant Networks

Cong Liu† Jie Wu‡ Xin Guan* Li Chen†
†Sun Yat-sen University ‡Temple University *Keio University

***Abstract*—Peer-to-peer file sharing techniques are efficient and popular ways to distribute network media contents. Aiming to improve the accessibility of media contents from mobile devices, we introduce peer-to-peer file sharing technique into delay tolerant networks (DTNs), which are formed solely by highly mobile devices and which distribute files utilizing the mobility of humans and vehicles. We first investigate two important implementation issues in a DTN environment: *cooperative file discovery* and *cooperative file download*. We then propose several distributed file discovery and file download protocols under different levels of user cooperation. Finally, we conduct extensive simulations using both real and synthetic DTN traces to evaluate the proposed protocols.**

***Index Terms*—Cooperative file sharing, delay tolerant networks (DTNs), metadata, simulation.**

I. Introduction

Mobility of humans and vehicles can be leveraged to enhance the network capacity and reachability. Numerous prior research work has been focused on *delay tolerant networks* (DTNs) [1], [2] where messages are forwarded in a mobility-assisted, store-carry-forward paradigm without the support of any infrastructure. In this paper, we study a very practical problem: introducing cooperative file sharing into DTNs. In our proposal, files can be downloaded from the Internet and cellular networks to mobile devices, such as laptops and cell phones. Moreover, they can also be requested and downloaded within the DTN formed solely by these cooperative mobile devices. This provides a more flexible, cheaper, and faster way to obtain media data than strictly Internet and cellular network download. Here, the DTN is an extension of the Internet for the more flexible access of files. We assume that files are generated on the Internet and they are downloaded by the nodes in the DTN. We name such a combination of the Internet and DTN a *hybrid DTN*. An example of the hybrid DTN is shown in Figure 1, where node 1 is downloading some files from the Internet and the other nodes in the DTN can download the file at a later time when they are connected to node 1.

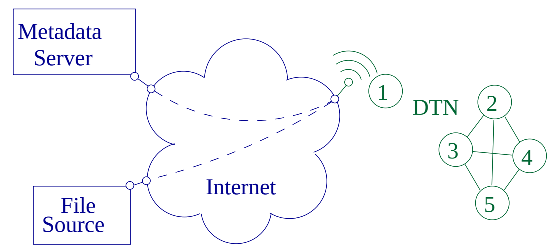

Fig. 1. An example hybrid DTN: node 1 downloads files from the Internet and shares the files as it connects to the other nodes in the DTN.

Our main thrusts in this paper are the following: (a) we envision a practical application scenario, in which cooperative file sharing is introduced in hybrid DTNs, and (b) we first investigate two important issues in the implementation of file sharing in hybrid DTNs, i.e. cooperative file discovery and file downloading.

In our cooperative file discovery, any user, including those who cannot access the Internet directly, can find the *uniform resource identifier* (URI) of the right file he/she needs using a keyword search. We enable file discovery in the DTN through the distribution of *metadata* (specifically, file description) among the DTN nodes. While different content distribution algorithms are proposed in prior research work, they assume that the nodes have the URIs of the contents they desired [3], [4]. In other algorithms [5], [6], [7], metadata is carried by a node only if the corresponding content is also stored in the same node. In their work, the purpose of carrying metadata is to confirm the identity of a content before downloading, but not for URI searching. Differently, we don't assume a one-to-one mapping of queries and metadata, and we allow metadata to be distributed within the DTN and to be discovered. Metadata acts as advertisement of files, and they can be distributed even before the files are produced.

Manual metadata selection can be a very helpful step in file discovery. Users of peer-to-peer file sharing software might notice that, sometimes, it is very difficult to choose the right metadata, such as a bit-torrent file, to download the content with. The reasons can be: (a) there are fake files, files with inferior quality, and different files with similar names, and (b) choosing an unpopular file (among the right files) will significantly prolong the download time. Therefore, before the download of a file, we first discover its metadata with optional user interventions. The advantages of using metadata are: (a) metadata use little bandwidth because they are much smaller than files, and the abundance of metadata enables users to the choose right files, which prevents redundant downloads, and (b) with the knowledge of what the users want, cooperative file downloading, the other issue we investigate, can be more efficient.

To the best of our knowledge, all existing DTN content distribution algorithms use pair-wise transmission between nodes. However, the broadcast nature of wireless transmission makes pair-wise transmission inefficient, especially in the rendezvous of DTN nodes where network density is high. The reasons are: (a) pair-wise transmission introduces contention between geometrically close transmission links, and (b) transmission efficiency is low because there is exactly one receiver in each transmission. We propose a broadcast-based file download algorithm. In this algorithm, for a set of nodes that can receive from each other, only one of them is allowed to send at a time, during which all other nodes are silent receivers. Nodes in such a set need to cooperatively determine their transmission order according to the importance of the metadata and the files they have. We show theoretically that the broadcast-based file download has an increasing per-node transmission capacity as node density increases. Meanwhile, the per-node transmission capacity of the pair-wise file download decreases as density increases.

Based on the proposed cooperative file discovery and file download schemes, we propose different metadata discovery and file download protocols under different levels of user cooperation. We evaluate the proposed protocols using simulations driven by both real and synthetic DTN traces. In these simulations, we compare the proposed protocols to simpler metadata discovery or file download protocols to provide insights into the relation between performance and design in the file download systems in hybrid DTNs and to verify the efficiency of the proposed protocols.

The remainder of this paper is organized as follows: Section II brings us into the context of DTNs and peer-to-peer file sharing systems. Section III describes the network model and an overview of our contributions with a motivating application scenario. In Sections IV and V, we present the proposed cooperative file discovery and file download protocols, respectively. In Section VI, we show our simulation methods and results. We conclude in Section VII.

II. Related Work and Challenges

A. Delay Tolerant Networks

Delay tolerant networks (DTNs) [1], [2] are occasionally-connected networks that suffer from frequent network partition. The Delay Tolerant Network Research Group (DTNRG) [2] has designed a complete architecture to support various protocols in DTNs. A DTN can be described abstractly using a space time graph [8] in which each edge corresponds to a contact. A *contact* is a period of time during which two nodes can communicate with each other. On the Internet, intermittent connectivity causes loss of data, whereas DTNs support communication between intermittently-connected nodes using the *store-carry-forward* routing mechanism. Routing (sending a message from one node to another) in the DTN is an active research area. Numerous routing protocols have been proposed [1], [8], [9], [10]. On the other hand, file discovery and file download in DTNs is a new area.

B. Peer-to-peer file sharing: BitTorrent

The BitTorrent protocol [12], [13], [14], [15], [16] focuses on bulk data transfer. *Metadata* specifies the name and the size of the file to be downloaded, as well as SHA-1 checksums of the data blocks. The metadata file also specifies the address of a *tracker* server for the torrent, which coordinates the interactions between the peers. The tracker maintains a list of currently active peers and delivers a random subset of these to clients, upon request. A BitTorrent peer uses a rate-based *tit-for-tat* (TFT) strategy to determine which peers to include in the set of *unchoked* peers to which it sends data. Each round, a peer sends data to unchoked peers from which it received data most rapidly in the recent past. This strategy is intended to provide incentives for contributing to the system and inhibit *free-rider* (peers download content without uploading to other peers).

C. Content Distribution in Delay Tolerant Networks

Communication methods in the DTN include: unicast, broadcast [17], multicast [3], and content distribution [5], [4], [6], [7]. Content distribution is the closest to our work. DTN content distribution systems extend the coverage of its counterpart in classical local area wireless networks by enabling content transfers between mobile nodes in an ad hoc network manner. The content distribution systems (e.g. podcasting distribution systems [3]) are usually receiver driven, in which nodes solicit contents of interest. The most significant difference between our DTN file sharing system and the previous content distribution systems is that there is a file discovery step in which the URIs of the contents are discovered by the nodes. This step makes our system more practical because we do not have assumption that the nodes are aware of the URIs of the file or contents in advance. Moreover, since existing content distribution algorithms use pair-wise communication to simplify implementation, which is not efficient for content distribution protocols in wireless environments, we propose a broadcast-based file download algorithm.

D. Challenges in a DTN Cooperative File Sharing System

The challenges of a cooperative file sharing system are in both file discovery and file download. File discovery (searching for metadata) is unlike searching on the Internet, which can be implemented centrally. To the best of our knowledge, file discovery (metadata searching) in the DTN (rather than sending queries to the Internet via DTN nodes) has not been investigated before, and it is challenging to enable efficient searching on DTN nodes, which can be isolated from the Internet. We enable searching through the distribution of metadata of files. We also provide a tit-for-tat algorithm to encourage every node to contribute to the discovery and distribution of metadata. File download in wireless environments is, on one hand, simpler than that on the Internet in the aspect that it is not necessary to determine the set of unchoked peers. On the other hand, it is also challenging to design an efficient broadcast-based file download algorithm which provides incentives for contribution to the system and inhibits free-riders.

III. Overview

A. Hybrid Delay Tolerant Network

A hybrid DTN is a DTN that surrounds the Internet. When a mobile node (wireless device) in the hybrid DTN is connected to the Internet, e.g., through a free WIFI access point, it can function as a normal computer and download data files from file servers on the Internet. Nodes in the hybrid DTN can also communicate directly to share files with each other.

We consider a scenario in which the Internet is the sole source of files, and the nodes in the hybrid DTN are the file downloaders. Nodes that have opportunity to connect to the Internet download files directly from the Internet. The nodes that have no Internet access can also download files with the help of other nodes in the hybrid DTN.

We focus on the cooperation of the nodes in the hybrid DTN. A hybrid DTN functions as an extension of the Internet, which effectively provides the hybrid DTN nodes with a wider and faster accessibility to the files on the Internet. We will refer to a node in the hybrid DTN simply as a node in the rest of the paper. As Internet access is ubiquitous, our cooperative file download application in the hybrid DTN may put some part of it in a central server on the Internet for better performance in terms of storage, computation, and accessibility.

B. Cooperative File Sharing in Hybrid DTNs

Files (video or audio clips) are produced by well known organizations or companies, such as FOX and ABC. Large files are divided into pieces of 256KB. Each file is associated with a metadata that contains information about the file including (a) the file name, (b) the file publisher, (c) the file description, e.g. an advertisement with a poster picture, (d) the uniform resource identifier (URI) of the file, (e) the checksums of its pieces, and (f) authentication information of the metadata against fake publishers. Metadata can be placed on different servers than those of their files on the Internet. The size of the pieces can be increased if we want to decrease the size of metadata, each of which contains the checksums of all pieces of a file.

Different from the metadata in BitTorrent, our metadata contain more information for the users of the nodes to determine which file to download. Each node runs a file discovery process and a file download process. The file discovery process collects metadata and stores them in the local storage of the node. When a user wants to search for a file, he or she inputs a query string and the file discovery process running in the node returns a sorted list of matched metadata and displays their information in a preferential order to the user. After examining this information, the user may select one of the metadata and download the corresponding file. The pieces of a file, which are stamped with the URI of the file and different offsets in the file, may be downloaded at different times and places.

We believe that such a cooperative file sharing application has a great potential for wide adoption, as it is cheaper than 3G cellular service and it is faster than downloading only using free WIFI. It is valuable and important to find an implementation of this cooperative file sharing system, which provides a responsive file discovery and fast file download.

To the best of our knowledge, all previous work in content distribution uses pair-wise wireless transmission. In our system, all communications use broadcasts to improve efficiency. Messages exchanged among the nodes include: (a) hello messages, (b) metadata, and (c) file pieces. Nodes send hello messages at least every second. A hello message includes the following information: (a) node ID, (b) the IDs of the nodes from which hello messages were received in the past 5 seconds, (c) query strings, and (4) the URIs of the downloading files. From the hello messages received, each node knows the neighbor nodes who can receive its messages. The metadata, query strings, and requesting URIs received form other nodes are stored.

IV. File Discovery

Cooperative file discovery is one of our major contributions, through which we first enable searching in a delay tolerant environment. The goal of the file discovery process is to download metadata that matches the user query strings and, probably, the metadata that will match future queries.

This file discovery scheme differs from existing content distribution schemes in that it separates the distribution of metadata and files. In our scheme, metadata are distributed earlier, in larger amounts, and are stored in the nodes for longer durations than files. Considering the possibly huge amount of metadata on the Internet, it is impossible to broadcast all metadata in the hybrid DTN. Metadata is distributed using a popularity-based method through both the pull and push methods.

The pull method works as follows: when a node is connected to the Internet, it sends query strings it has to the metadata server on the Internet, which returns the best matched metadata. When a node is connected to other nodes, it pulls metadata in a similar way. Since each user may only generate a small number of query strings, nodes can also store the query strings of their most frequently connected nodes to cooperatively shorten file discovery time.

After sending metadata to a peer node, which pulls the metadata, a node can continue to push to the peer node other metadata. The pull-based metadata distribution is based on the popularities of the metadata, which can be calculated from a central server on the Internet.

A. Cooperative File Discovery

We first consider the cooperative (or altruistic) case where the nodes try their best to send metadata that match other nodes' queries. Since the opportunistic connections may stop at any time as the nodes move away from each other, nodes need to send the metadata that match more queries of other nodes first.

The popularity of a metadata is defined based on the number of users requesting to download the corresponding file of the metadata after the users viewing the information contained in the metadata. Assuming a positive trend, the more popular a metadata was in the past, the more likely it will be selected by other users in the future.

The popularity of a metadata is the popularity of the metadata in the whole network. The popularities, which ranges from 0 to 1, can be maintained by a central metadata server. One way to define the popularity of a metadata could be the percentage of Internet access nodes requesting the file of the metadata in the past 24 hours. The popularities of metadata can be downloaded from the server and updated to the nodes as they connect to the Internet.

Each node sends metadata in two phases. In the first phase, metadata that match the query strings of the connected nodes are sent. Those that match the query strings of more nodes themselves are sent. In this phrase, metadata that match the same number of query strings are sent in the order of decreasing popularity. In the second phase, other metadata that do not match any queries are sent in the order of decreasing popularity.

B. Tit-for-Tat File Discovery

Second, we consider the selfish case where some of the nodes do not want to contribute to the file discovery process. Unlike the tit-for-tat used by BitTorrent clients, our tit-for-tat algorithm does not choke peers because the wireless communication is broadcast in nature[1]. We weigh metadata by the sum of the *credits* of the nodes requesting the metadata. Each node u maintains a credit value for each other node v. The credit of v is proportional to the number of the metadata that u received from v that u requested. For example, if v sends to u a new metadata that matches some of u's query strings, then v's credit is increased by 5; otherwise, if v sends to u a new metadata that u is not interested in, then v's credit is increased by the popularity of the metadata.

A node that contributes more can have its credit higher. Consequently, its queries are weighed more, and it is likely to receive its desired metadata earlier. Receiving the desired metadata faster is necessary to facilitate a node's file download. Under such a tit-for-tat algorithm, if a node wants other nodes to send metadata that match its queries, the node should send metadata that match other nodes' queries and those with high popularities.

Note that due to the broadcast nature of wireless networks, free-riders cannot be completely inhibited. A free-rider may still have a chance to receive the metadata it needs without sending anything to other nodes. However, when the amount of different metadata is large, the chance of receiving a desired metadata among randomly received metadata is small. Therefore, our tit-for-tat algorithm provides an incentive for the cooperation of the nodes in the file discovery process.

V. Broadcast-based File Download

It is well known that in a pedestrian mobile network, the majority of connections are short. Therefore, it is reasonable to let the file discovery process use the starting period of time in each connection: (a) file discovery can be finished in short connections since the size of metadata is small, (b) the efficiency of the pull-based file discovery is more related to the number of nodes met than to the length of each connection, (c) short connections are less useful for downloading bulky file pieces, and (d) we can assume that if the connection is still there after the file discovery process exchanging metadata, the connection is likely to be a long one.

When the file download process uses long connections, structures can be built among the nodes to facilitate cooperation and reduce collision. Our broad-based file download process divides nodes into cliques, in which each node can receive messages from each other. Since each node periodically sends hello messages, which contain the set of IDs of other nodes from which the node can receive messages, each node can calculate all the maximum cliques containing it. For simplicity, we do not consider interference between cliques. A broadcast-based communication algorithm is particularly efficient in a file download. It is not difficult to find that per node communication bandwidth for broadcast-based communication is $\frac{n-1}{n}$ for n nodes, while it is $\frac{1}{n}$ for pair-wise communication.

A. Cooperative File Download

We first consider the altruistic case, where the nodes cooperatively send file pieces requested by other nodes. To prevent collisions and facilitate cooperation, a coordinator is selected in each clique. The coordinator determines the order in which file pieces are broadcasted in the clique.

Each clique of nodes send file pieces in two phases. In the first phase, file pieces requested by the nodes in the clique are sent. Those requested by more nodes are sent first. File pieces requested by equal numbers of nodes are broadcast in decreasing file popularity. In the second phase, other file pieces are sent in decreasing popularity.

B. Tit-for-Tat File Download

Second, we consider the selfish case where some of the nodes do not want to contribute to the file downloading process. In this case, it is not suitable to select a coordinator to control the communication because a selfish coordinator may instruct to broadcast the messages which only the selfish coordinator itself is interested in. We allow nodes to broadcast in turn in an agreed-upon cyclic order. For example, such an order can be generated by a pseudo random number generator known by all nodes, using the sum of the IDs as the seed for the random generator.

Our tit-for-tat algorithm for file downloads uses the same credit mechanism that is used in the tit-for-tat algorithm for file discovery. To gain reciprocation from other nodes, each node will try send the file pieces requested by other nodes and those with high popularities.

VI. Simulation

A. Simulation Method

For evaluation purposes, we implement a simplified version of the proposed protocol, in which we assume all nodes are cooperative. We use the UMassDieselNet trace [18], [19] and the National University of Singapore (NUS) student trace [20] to perform simulation evaluations. The assumption that communication cliques do not overlap is true in these traces

[1]Peers can still be choked if encryption is used. We will leave this topic for future work.

we use: (a) the UMassDieselNet trace only contains pair-wise contacts, and (b) in the NUS student trace, students can receive messages from each other if and only if they are in the same classroom.

Our simulation model is specified as follows: not all nodes can access the Internet. The nodes that can access the Internet are called *Internet access nodes*. The percentage of Internet access nodes ranges from 10% to 90%. We allow all Internet access nodes to have enough bandwidth to download the files they need, and we only measure the delivery ratio (of metadata and files) of the non-Internet access nodes.

A number n (ranging from 10 to 100) of new files are generated on the Internet everyday at 12PM, all of which have a time-to-live (TTL) (ranging from 1 to 5 days). Each file is generated with a popularity p, which is the probability that each node is interested in this file. At the same time, each node generates queries for these new files according to their popularity. We simply assume the probability density function of file popularity to be $\frac{\lambda e^{-\lambda x}}{1-e^{-\lambda}}$, and we can use $p = -\frac{\log(1-x(1-e^{-\lambda}))}{\lambda}$ to generate the file popularities, where x is a random variable distributed uniformly over 0 to 1. The means of p is approximately $\frac{1}{\lambda}$. We set $\lambda = \frac{n}{2}$, and therefore, the average number of queries each node generates per day is $np \approx \frac{n}{lambda} = 2$.

We assume that in each contact, nodes can send or receive a fixed number (ranging from 1 to 10) of metadata and files. Metadata and files requested by the nodes in a clique are sent first, followed by files. Nodes determine their frequent contacting nodes from statistics of the traces. In the UMassDieselNet trace, nodes that have contacts at least every three days are frequent contacting nodes; in the NUS student trace, nodes that have contacts at least once per day are frequent contacting nodes.

We compare the proposed protocols to simpler metadata or file distribution protocols to provide insights into the relation between performance and design in the DTN file download systems and to verify the efficiency of the proposed protocols. We name the proposed system *mobile BitTorrent* (MBT). We compare three protocols: (a) MBT, (b) MBT-Q (without distribution of queries), and (c) MBT-QM (without distribution of both queries and metadata). In MBT-Q, a node can only pull metadata from other nodes, but it cannot ask its frequent contacting nodes to collect the metadata it is interested in. Similarly, in MBT-QM, a node can only pull files from other nodes.

We use real traces, the *UMassDieselNet trace*, and synthetic traces, the *NUS student trace*, in our simulations. For space limitation, we omit the details of these traces. For interested readers, please refer to our previous work [21].

B. Simulation Results

In our simulation results, the performance measurements we use are delivery ratios of metadata and files, which is the ratio of the number of delivered metadata and files over the total number of queries generated. Performance is measured among the non-Internet access nodes.

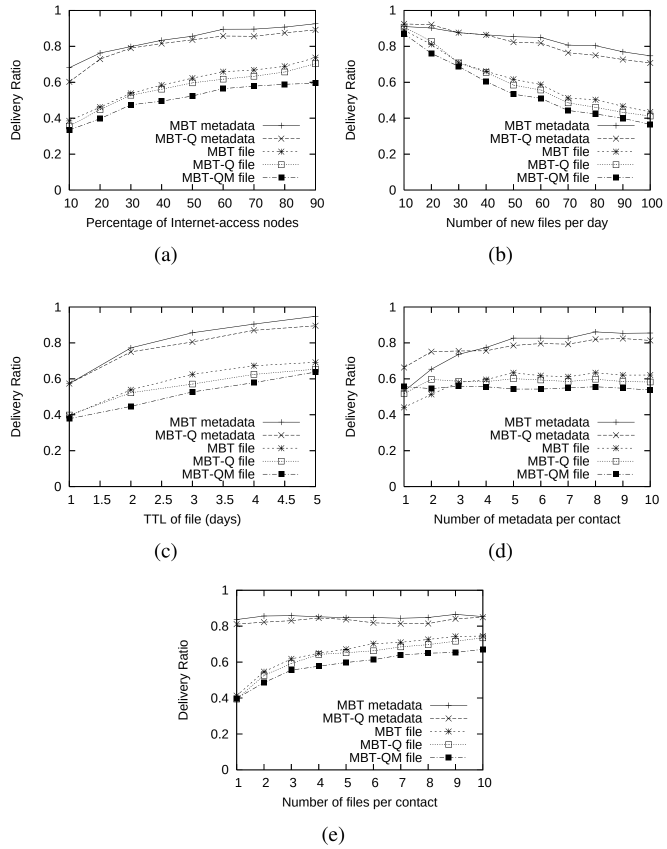

Fig. 2. Performance in UMassDieselNet trace.

UMassDieselNet trace: As shown in Figures 2(a) to 2(e), the delivery ratios of both metadata and files, for all protocols, increase as (a) the percentage of Internet access nodes increases, (b) the number of new files generated per day decreases, (c) the time-to-live of the file increases, (d) the number of metadata exchanged per contact increases, and (e) the number of files exchanged per contact increases. Besides an exceptional case, the delivery ratio of MBT is the best and that of MBT-QM is worst. This exception happens when the number of metadata exchanged per contact is very small (Figure 2(d)). We believe that the reason for this is that: when the number of metadata is small, the requests represented by these metadata are biased. This makes a global popularity-based MBT-QM get more files that are requested by the nodes. The same explanation can be applied to the question of why MBT-Q has a higher metadata delivery ratio.

NUS student trace: As shown in Figures 3(a) to 3(f), the results in the NUS student contact trace are similar to those in the UMassDieselNet trace. Additionally, as shown in Figure 3(a), the file delivery ratio of MBT and MBT-Q increases very fast as the percentage of Internet access nodes increases; meanwhile, MBT-QM shows no increase because it does not have a file discovery process. This shows the efficiency of our file discovery process: with 80% of Internet access nodes, the file delivery rate doubles when using file

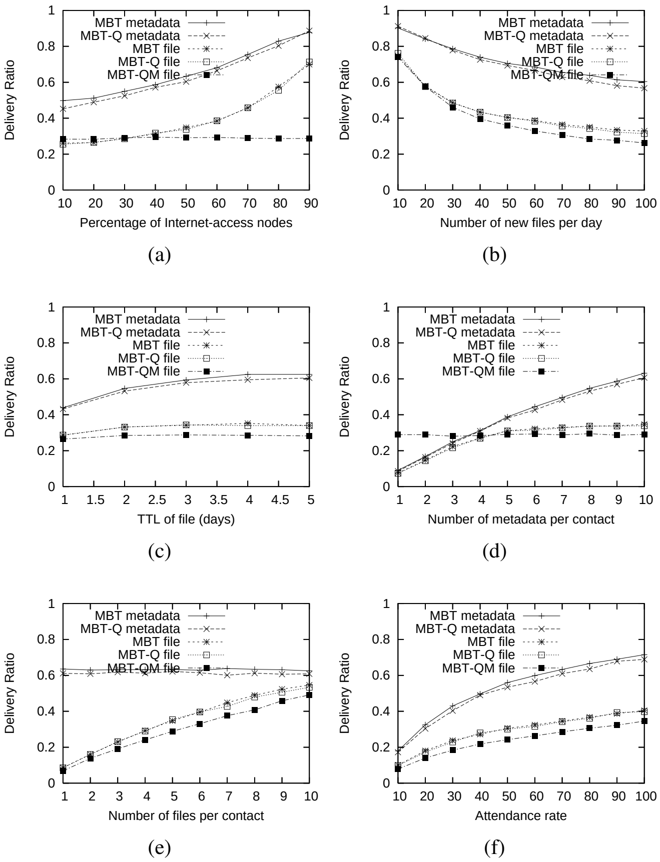

Fig. 3. Performance in NUS student trace.

discovery.

VII. Conclusion

In this paper, we improved the accessibility of files from mobile devices with a cooperative file sharing system in DTNs, which is formed solely by mobile devices and distributes files utilizing the mobility of humans and vehicles. We envisioned a practical application scenario of this cooperative file sharing system in DTNs, and proposed several protocols for file discovery and file download. Finally, we conducted extensive simulations using both real and synthetic DTN traces to evaluate the proposed protocols. Simulation results indicate the efficacy of the proposed protocols. Our future work will consider more complicated simulation models and the deployment of our protocol on real devices.

References

[1] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. of ACM SIGCOMM*, 2004.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant Network Architecture. In *Internet draft: draft-irrf-dtnrg-arch.txt, DTN Research Group*, 2006.

[3] W. Zhao, M. Ammar, and E. Zegura. Multicasting in Delay Tolerant Nnetworks: Semantic Models and Routing Algorithms. In *Proc. of ACM SIGCOMM DTN Workshop*, 2005.

[4] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla. Bluetorrent: Cooperative content sharing for bluetooth users. In *Proc. of ACM PERCOM*, 2006.

[5] J. Leguay, A. Lindgren, T. Friedman, J. W. Scott, , and J. Crowcroft. Opportunistic Content Distribution in an Urban Setting. In *Proc. of ACM CHANTS*, 2006.

[6] A.-K. Pietilainen and C. Diot. Social Pocket Switched Networks. In *Proc. of IEEE INFOCOM*, 2009.

[7] G. Sollazzo, M. Musolesi, and C. Mascolo. TACO-DTN: A Time-Aware COntent-based dissemination system for Delay Tolerant Networks. In *Proc. of ACM MobiOpp*, 2007.

[8] S. Merugu, M. Ammar, and E. Zegura. Routing in Space and Time in Network with Predictable Mobility. In *Technical report: GIT-CC-04-07, College of Computing, Georgia Tech*, 2004.

[9] J. Haas, J. Y. Halpern, and L. Li. Gossip-Based Ad Hoc Routing. In *Proc. of IEEE INFOCOM*, 2002.

[10] A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *Lecture Notes in Computer Science*, 3126:239–254, August 2004.

[11] C. Liu and J. Wu. An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks. In *Proc. of ACM MobiHoc*, 2009.

[12] B. Cohen. Incentives build robustness in BitTorrent. In *Proc. of IEEE IPTPS*, 2003.

[13] B. Fan, D.-M. Chiu, and J. Liu. The Delicate Tradeoffs in BitTorrent-like File Sharing Protocol Design. In *Proc. of IEEE ICNP*, 2006.

[14] A. Bharambe, C. Herley, and V. Padmanabhan. Analyzing and Improving a BitTorrent Networks Performance Mechanisms. In *Proc. of IEEE INFOCOM*, 2006.

[15] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do Incentives Build Robustness in BitTorrent. In *Proc. of USENIX NSDI*, 2007.

[16] A.L.H. Chow, L. Golubchik, and V. Misra. BitTorrent: An Extensible Heterogeneous Model. In *Proc. of IEEE INFOCOM*, 2009.

[17] M. May, C. Wacha, V. Lenders, and G. Karlsson. Wireless Opportunistic Podcasting: Implementation and Design Tradeoffs. In *Proc. of ACM CHANTS*, 2007.

[18] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In *Proc. of IEEE INFOCOM*, 2006.

[19] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing. In *Proc. of ACM MobiCom*, 2007.

[20] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and Implications of Student Contact Patterns Derived from Campus Schedules. In *Proc. of ACM MobiCom*, 2006.

[21] C. Liu and J. Wu. Routing in a Cyclic MobiSpace. In *Proc. of ACM MobiHoc*, 2008.