A Peer-to-Peer Approach for Mobile File Transfer in Opportunistic People Networks*

Ling-Jyh Chen and Ting-Kai Huang

Institute of Information Science, Academia Sinica
{cclljj, tkhuang}@iis.sinica.edu.tw

Abstract

With wireless technologies extending to every part of our daily lives, mobile networking applications are becoming increasingly popular for accessing the Internet. In this paper, we propose a peer-to-peer approach, called M-FTP, for mobile FTP services. Unlike traditional approaches, the proposed scheme implements a Collaborative Forwarding *algorithm that takes advantage of opportunistic wireless connections, thereby improving network capacity by exploiting the diversity of network mobility. Using simulations as well as real-world network scenarios, we demonstrate that the proposed scheme provides a better mobile FTP service than traditional schemes, and thus facilitates more effective data transfer on the go.*

1 Introduction

The last few years have seen an impressive growth in computer network applications. One striking success in this area has been the Internet, which has successfully accelerated the dissemination of information and knowledge by overcoming geographic boundaries. Wireless technologies represent another orthogonal area of growth: in both wide area applications like 2.5G/3G and local area applications like 802.11b/g and Bluetooth. As wireless technologies continue to extend into every part of our working and living environments, it is becoming increasingly desirable to have a solution that can provide effective data transfer on the go.

Proper handling of mobility is the key to the success of mobile networking applications, and a great amount of research effort has been invested to facilitate mobile applications in the last few years [8, 14, 22, 23, 28, 29, 31, 33]. Generally, these approaches can be categorized into three types: network layer-based [14, 28], transport layer-based [8, 23, 33], and proxy-based [22, 29, 31]. The common goal of these approaches is to maintain the Internet connectivity for mobile users even when they perform vertical handoffs between different networks. However, the major shortcoming of these approaches is that they are all Internet-based and thus limited in providing FTP service when the Internet connection is not always available.

Fortunately, as recent studies have reported that intermittent network connection is inevitable for mobile users on a daily basis [10, 11], Grossglauser et al. [16] have also shown that network capacity can be increased dramatically by exploiting node mobility as a type of multi-user diversity. In other words, opportunistic ad hoc connections can be useful for extending the coverage of wireless communications.

Several approaches have been proposed to allow mobile users to request and transfer data on the go [6, 18, 20, 24, 25, 26, 27, 32]. The approaches are either cache-based or Infostation-based. Cache-based approaches facilitate mobile file transfer by prefetching popularly requested files (e.g., commercial Ads, movie trailers, and song previews) to a local storage. Infostation-based approaches, on the other hand, support on-demand FTP requests by deploying dedicated servers as bridges between the Internet and mobile networks. However, the capability of these approaches for mobile file transfer is limited because they are basically centralized and fail to exploit the diversity of network mobility.

In this study, we propose a peer-to-peer approach, called M-FTP, for mobile file transfer applications. In addition to combining the strengths of cache-based approaches (i.e., prefetching the most likely requested files to a local storage) and Infostation-based approaches (i.e., allowing on-demand FTP requests), the

[*] This work was partially supported by the National Science Council of Taiwan under grant numbers NSC 96-2623-7-001-004-D and NSC 96-2221-E-001-010.

M-FTP scheme implements a *Collaborative Forwarding* algorithm to further utilize opportunistic ad hoc connections and spare storage in the network. Using simulations as well as real-world mobility traces, we evaluate the proposed scheme in terms of service ratio and traffic overhead. The results show that the scheme significantly outperforms previous approaches in all test cases, while its traffic overhead remains moderate.

The remainder of this paper is organized as follows. In Section 2, we review related works on mobile file transfer in mobile networks. In Section 3, we describe the M-FTP scheme and the *Collaborative Forwarding* feature used to disseminate data in mobile networks. Section 4 presents a comprehensive set of simulation results, which we analyze and explain in detail. We then present our conclusions in Section 5.

2 Related Works

Data transfer in mobile networks has been researched for a number of years, and several approaches have been proposed [6, 18, 20, 24, 25, 26, 27, 32]. The approaches can be roughly classified into two types: cache-based and Infostation-based approaches.

Basically, cache-based approaches automatically download those files that are considered likely to be requested in the near future. This is done in either a *push*-based [6, 25, 32] or a *pull*-based [20, 27] fashion. More precisely, when *push*-based approaches are used, the content provider automatically supplies a mobile user with popularly requested files as long as he/she is connected to the Internet and has free storage space. In contrast, under *pull*-based approaches, the mobile device automatically pulls (pre-fetches) files (using its own content selection algorithm) without the FTP requests issued manually by mobile users. There is a trade-off in these approaches that they incur a tremendous storage overhead in return for the performance gain (i.e., the more files they cache in the local storage, the greater the likelihood that they will be able to serve the next FTP request without consuming extra Internet bandwidth). However, this is considered infeasible for emerging power/storage-constrained handheld devices. In addition, these approaches only allow Internet-capable users to download files, but they do not provide a way for Internet-incapable users.

Infostation-based approaches provide Internet access for mobile users by installing Infostations to act as bridges between the Internet and mobile networks [15, 30], such as Mobile Ad hoc NETworks (MANET) or Delay/disruption Tolerant Networks (DTN). The advantage of such approaches is that they allow mobile users to obtain/publish data on the Internet via Infostations by using local wireless connections (e.g., WiFi and Bluetooth). Hence, mobile file transfer is possible, even when mobile users are not directly connected to the Internet. For example, [18] proposes a solution called *Mobile Hotspots*, which provides mobile Internet access in railway systems; [24] proposes the *SPAWN* scheme that provides cooperative content delivery via Internet gateways to the inner vehicular ad hoc networks (VANET); and [26] proposes a Bundle Routers (BR)-based approach that deploys BRs as Infostations to separate the Internet and DTNs. Unlike previous schemes, the SPAWN scheme and the BR-based scheme require mobile users' collaboration to forward data in a multihop and store-carry-and-forward fashion. It also allows users to access Internet data, even if they cannot locate an Infostation. However, the major shortcoming of the above approaches is that they require dedicated Infostations (i.e., gateways or BRs), which act as gateways to the Internet; thus, they suffer from scalability and single-point-of-failure problems.

3 The Proposed Approach: M-FTP

In this section, we present our peer-to-peer approach, called M-FTP, for mobile file transfer in opportunistic people networks. The approach combines the strengths of cache-based and Infostation-based approaches with peer-to-peer networking concepts. As a result, it is better able to cope with the intermittent network connectivity caused by mobility, and can therefore provide better mobile FTP services. We describe M-FTP in detail in the following sub-sections.

3.1 M-FTP Architecture

There are two types of participating peers in the M-FTP system: *Gateway Peers* (GP) and *Vanilla Peers* (VP). GPs are connected to the Internet directly (by using, for example, GPRS, UMTS, WiMAX, or WiFi/Bluetooth via Internet Access Points). VPs are peers that do not have Internet access, but they have local wireless connection capabilities (by using, for example, WiFi, Bluetooth, or infrared via the ad hoc connection mode). Note that a mobile peer may switch from the GP mode to the VP mode (and vice versa) if it temporarily loses (or recovers) its Internet connection (e.g., when entering/leaving an elevator or a tunnel).

In the M-FTP system, there are two cases when a peer A requests to download a file: if A is a GP, he can download the file himself immediately; otherwise (i.e., A is a VP), A forwards his request, with a replication factor f (i.e., f copies of the request are input to the

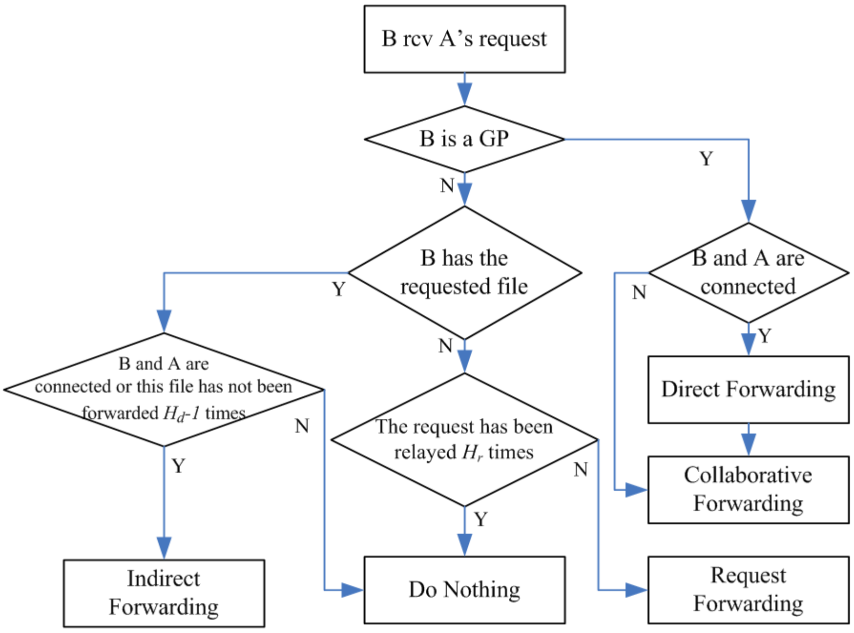


Figure 1. Illustration of the request process algorithm of the M-FTP scheme.

network), to the first f peers he meets in the network. Of course, the larger the value of f, the higher the number of participating peers that will be aware of A's request; however, the traffic and storage overhead also increase linearly as f increases.

The proposed M-FTP system is then applied as follows (see Fig. 1). Suppose B is another mobile peer that receives A's request. There are two cases:

1. *If B is a GP*, he immediately downloads the requested file from the Internet, and forwards the file to A if they are directly connected (i.e., by the *Direct Forwarding* algorithm). Next, B disseminates the file to the mobile network using the *Collaborative Forwarding* algorithm, which we will discuss in the next subsection. Note that the objective of the *Collaborative Forwarding* algorithm is to cache files previously requested by peers in the network. This allows the M-FTP system to reduce redundant downloading when multiple peers request the same file. Of course, proper buffer management is required to further improve the performance of the M-FTP system. We defer a detailed discussion and evaluation of this issue to a future work.

2. *If B is a VP*, he first checks his local storage to determine whether the requested file has been cached, and then implements one of the following two options:

   (a) *If B has the file requested by A*, he forwards it to A if they are directly connected or the file has not been forwarded $H_d - 1$ times (i.e., by the *Indirect Forwarding* algorithm); otherwise, he does nothing[1]. Note that the *Indirect Forwarding* phase is slightly different to the *Direct Forwarding* phase, since B may only have a portion of the requested file (which depends on the underlying *Collaborative Forwarding* algorithm); whereas the complete file is forwarded in the *Direct Forwarding* phase.

   (b) *If B does not have the file requested by A*, he forwards A's request to his next encountered peer so long as the request has not been forwarded (from A to B) more than H_r times (i.e., the *Request Forwarding* algorithm); otherwise, he does nothing.

Additionally, the M-FTP system prioritizes the data transmissions such that *Direct Forwarding* has the highest priority, followed by *Request Forwarding.* Then, the other types of transmissions are processed on a first-come-first-served basis. The reason is that *Direct Forwarding* can complete a FTP session, and thus shorten the service time of that transmitted page. Similarly, *Request Forwarding* propagates FTP requests to the network, which increases the probability that the

[1] In order to reduce the overall traffic overhead in the network, each data block is allowed to be forwarded at most H_d times in the M-FTP scheme.

requests will reach GPs and thus be cached in the network.

3.2 Collaborative Forwarding Algorithm

In this study, we incorporate the Probabilistic Routing (PROPHET) algorithm [21] to provide collaborative forwarding for the M-FTP system. The PROPHET scheme is based on the *Epidemic Routing* scheme [34], where each node blindly floods blocks of data to its neighbors as long as they are within its transmission range. In contrast, the PROPHET scheme forwards data blocks to a newly encountered node only if that node has a higher likelihood of successfully forwarding the message (called the *delivery predictability*) to the destination peer than the current peer.

More specifically, the PROPHET scheme is designed based on an assumption that *the node that is often encountered has a higher delivery predictability than the others*. Thus, when peer A encounters peer B, the PROPHET scheme immediately updates the corresponding *delivery predictability* ($P_{A,B}$) using Eq. 1, where $P_{encounter}$ is a constant and $0 \leq P_{encounter} \leq 1$.

$$P_{A,B}^{new} = P_{A,B}^{old} + (1 - P_{A,B}^{old}) \times P_{encounter}. \qquad (1)$$

Moreover, if A and B do not encounter each other during a period of time, they are less likely to be good forwarders to each other, and the PROPHET scheme gradually reduce the delivery predictability using Eq. 2, where K is the number of time units since the last time A and B were encountered, and γ is an aging constant ($0 \leq \gamma \leq 1$).

$$P_{A,B}^{new} = P_{A,B}^{old} \times \gamma^{K}. \qquad (2)$$

Additionally, the PROPHET scheme considers the *transitive property* when calculating the delivery predictability. More precisely, the delivery predictability $P_{A,B}$ comprises two parts, namely the direct message transfer (i.e., A transmits data to B directly) and the indirect message transfer (i.e., A has to rely on other nodes relaying data to B). The delivery predictability calculation is shown in Eq. 3, where C is the relay node, and β is a scaling constant determining the impact of indirect message transfer on the overall delivery predictability.

$$P_{A,B}^{new} = P_{A,B}^{old} + (1 - P_{A,B}^{old}) \times P_{A,C} \times P_{C,B} \times \beta. \qquad (3)$$

Note that the PROPHET scheme forwards data blocks to a newly encountered node only if that node has a greater *delivery predictability* to the destination peer than the current peer. Consequently, comparing to ordinary epidemic routing approach that blindly floods the network, the traffic overhead can be greatly reduced.

Table 1. Music types of the selected iTune songs

Music Type	# of songs	Music Type	# of songs
Pop	22	Soundtrack	9
Hip-Hop/Rap	29	R&B/Soul	6
Rock	12	Bandas sonoras	1
Alternative	10	iTunes Latino	1
Country	10		

4 Evaluation

We now evaluate the performance, in terms of the service ratio and traffic overhead, of the M-FTP scheme and the Mobile Hotspots scheme using a Java-based simulator, called DTNSIM [4]. As mentioned earlier, Mobile Hotspots is an Infostation-based approach with mobile Internet gateways, but without a collaborative forwarding feature (i.e., a mobile user can only download files if he encounters one of the gateways).

In each simulation run, we randomly select γ mobile peers as GPs (with unlimited Internet connection bandwidth) and 20% of the other peers (i.e., VPs) as FTP requesters. For simplicity, we focus on music downloading in this study; and to be realistic[2], the FTP requests are based on the distribution of the top 100 requested iTune [1] songs as reported in the iTune store on September 7, 2007. The 100 selected iTune songs are in Advanced Audio Coding (AAC) format [2], and the average file size is about 3.03MB. Table 1 shows the distribution of the music types of the selected songs.

We assume that FTP requesters only issue FTP requests in the first 10% of the simulation time, with a Poisson rate of 1,800 seconds/request. We also assume that data transmission between mobile peers is wireless at a fixed rate of 2Mbps, and that each FTP request can be relayed at most 2-hops far ($H_r = 2$). For the *Collaborative Forwarding* scheme (i.e., PROPHET), the $P_{encounter}$ parameter is set to 0.75, the aging constant γ is set to 0.98, the transitive property parameter β is set to 0.25, the buffer size of each mobile peer is unlimited, and each data block is allowed to be forwarded at most 5 times ($H_d = 5$). All the simulation

[2] We assume that the file that has been frequently downloaded in the history is more likely to be requested in the near future [7, 12].

Table 2. The properties of the two network scenarios

Trace Name	iMote	UCSD
Device	iMote	PDA
Network Type	Bluetooth	WiFi
Duration (days)	3	77
Devices participating	274	273
Number of contacts	28,217	195,364
Avg # Contacts/pair/day	0.25148	0.06834

results presented in this section are based on the average performance of 200 simulation runs.

4.1 Evaluation Scenarios

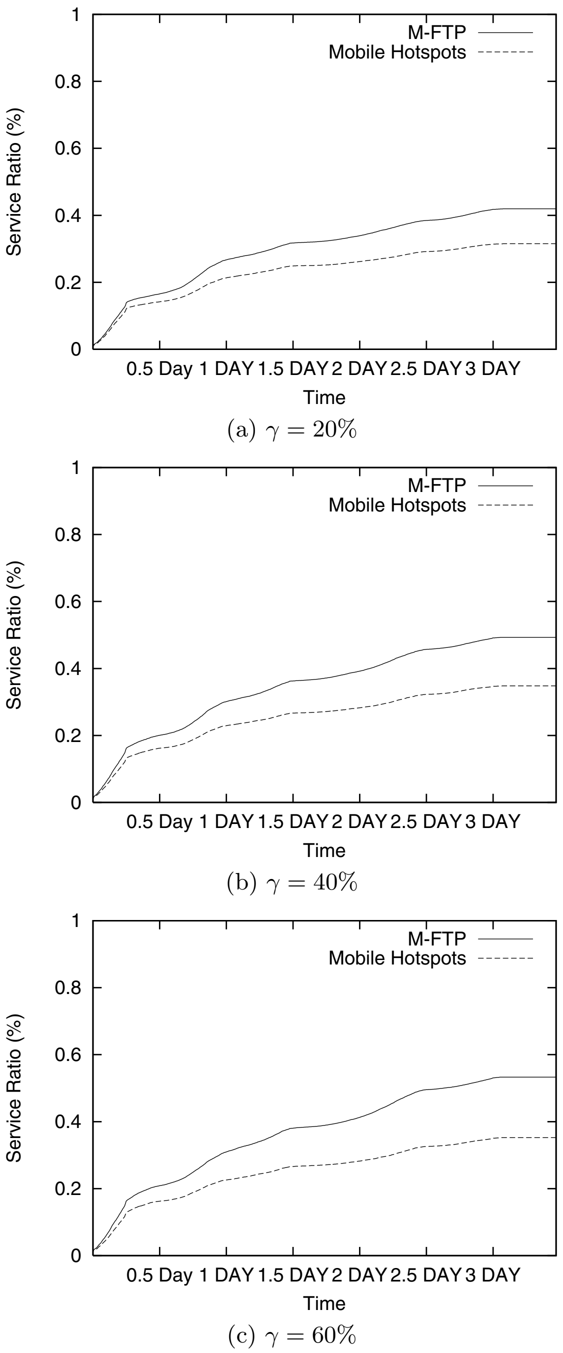
We evaluate two network scenarios based on realistic campus wireless network traces, namely, the iMote [3] and UCSD [5] traces, which are publicly available for research purposes. Table 2 outlines the basic properties of the network scenarios.

The UCSD trace is client-based and records the availability of WiFi-based access points (APs) for each participating portable device (e.g., PDAs and laptops) on the UCSD campus. The network trace covers a two and half-month period, and there are 273 participating devices. Similar to [11, 19], we assume that two participating devices in ad hoc mode encounter a communication opportunity (i.e., a network contact) if they are both associated with the same AP at the same time.

The iMote trace is a human mobility trace collected at the 2005 IEEE Infocom conference. It was aggregated from 41 Bluetooth-based iMote devices distributed to the student attendees for the duration of the 3-day conference. Each iMote device was pre-configured to periodically broadcast query packets to find other Bluetooth devices within range, and record the devices that responded to the queries. In addition to the distributed iMote devices, another 233 devices were recorded in the trace. They may have been other Bluetooth-enabled devices (e.g., PDAs, cell phones, or headsets) used during the conference. For simplicity, we assume there is a network contact between two Bluetooth devices if there are query-and-response interactions between them.

4.2 Evaluation I: Service Ratio

Here, we evaluate the service ratio performance of the M-FTP scheme and the *Mobile Hotspots* scheme. Figs. 2 and 3 show the experiment results with various γ values in the iMote and UCSD scenarios in Cumulative Distribution Function (CDF) curves.



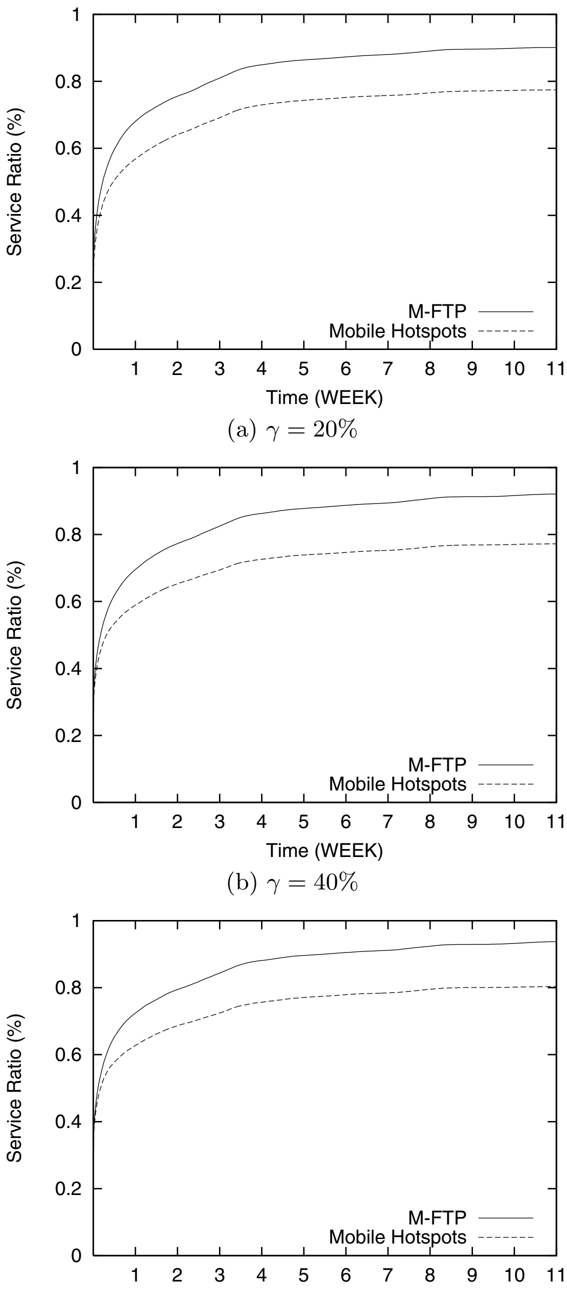
(a) $\gamma = 20\%$

(b) $\gamma = 40\%$

(c) $\gamma = 60\%$

Figure 2. Comparison of the service ratio performance of the M-FTP and Mobile Hotspots schemes with various numbers of GPs in the iMote scenario ($\gamma = 20, 40, 60\%$)

From Figures 2 and 3, we observe that the proposed M-FTP scheme outperforms the Mobile Hotspots scheme. More specifically, the M-FTP scheme is able



(a) $\gamma = 20\%$

(b) $\gamma = 40\%$

(c) $\gamma = 60\%$

Figure 3. Comparison of the service ratio performance of the M-FTP and Mobile Hotspots schemes with various numbers of GPs in the UCSD scenario ($\gamma = 20, 40, 60\%$)

to transmit more than 15% files in all test cases. This confirms our intuition that *collaborative forwarding* can utilize opportunistic connections, and thus better exploit the diversity of network mobility. We also observe that the service ratio improves as the value of γ increases, especially in the iMote scenario. The reason is that the iMote scenario's mobility trace was collected by 41 iMote devices, but it did not record network contacts among the other 233 external devices. Consequently, iMote devices generally have a much greater number of network contacts than external devices. Moreover, the Mobile Hotspots scheme only performs effectively if most GPs are iMote devices (otherwise, the GPs can not contribute much since they seldom encounter VPs in the network scenario). Thus, the higher the value of γ, the greater the likelihood that most GPs in the network will be iMote devices.

Table 3. Comparison of the traffic overhead performance of the M-FTP and Mobile Hotspots schemes (Units: Mbytes)

	γ	M-FTP	Mobile Hotspots	Normalized Overhead
iMote	20%	22,170	5,866	3.78
	40%	23,932	6,613	3.62
	60%	24,696	7,197	3.43
UCSD	20%	1,425,943	269,834	5.28
	40%	1,510,094	261,653	5.77
	60%	1,535,310	261,820	5.86

On the other hand, if each mobile peer has a similar number of network contacts (as in the UCSD scenario), the results show that the service ratio performance is consistent with the changes in the γ values for both schemes. Once again, M-FTP is superior to Mobile Hotspots in all test cases.

4.3 Evaluation II: Traffic Overhead

Next, we evaluate the traffic overhead of M-FTP and Mobile Hotspots. The simulation settings are the same as those in the previous subsection, and the results are based on the average traffic overhead of 200 simulations. Table 3 shows the results where the normalized overhead is derived by taking the ratio of the traffic overhead of the M-FTP scheme over the Mobile Hotspots scheme.

The results in Table 3 show that the normalized traffic overhead is about 3.5 in the iMote scenario and 5.5 in the UCSD scenario. The reason is because the M-FTP scheme disseminates FTP requests and downloaded files to multiple peers to take more aggressive advantage of the diversity of network mobility. Of course, the traffic overhead can be adjusted by tuning the parameters (i.e., H_r and H_d), replacing the collaborative forwarding algorithm, or implementing overhead reduction strategies (e.g., *Explicit ACKs* [9], *Passive*

Cure [17], *Time-To-Live* [17], *Kill Time* [17], and *Reverse Path Forwarding* [13]). Generally, the more replicated data stored in the network, the better the service ratio achieved by the M-FTP scheme. The tradeoff should be based on the required service ratio and the available buffer sizes. We defer a detailed discussion and evaluation of this issue to a future work.

5 Conclusion

In this study, we have proposed an approach called M-FTP to improve the effectiveness of FTP applications in mobile opportunistic networks. Unlike traditional approaches, M-FTP does not need dedicated servers to form bridges between the Internet and mobile networks because it is a peer-to-peer application. It implements a *Collaborative Forwarding* feature to make better use of opportunistic connections among mobile peers, and thereby improves the network capacity by exploiting the diversity of network mobility. Using simulations as well as real-world network scenarios, we evaluate M-FTP against a traditional approach called Mobile Hotspots. The results demonstrate that our scheme can achieve better service ratios with moderate traffic overhead in all test cases. Work on evaluating the proposed scheme with a buffer size constraint is ongoing. We will report the results in the near future.

References

[1] Apple - ipod + itunes. http://www.apple.com/itunes/.

[2] Apple - quicktime - technologies - aac audio. http://www.apple.com/quicktime/technologies/aac/.

[3] Crawdad project. http://crawdad.cs.dartmouth.edu/.

[4] Delay tolerant network simulator. http://www.dtnrg.org/code/dtnsim.tgz.

[5] Ucsd wireless topology discovery project. http://sysnet.ucsd.edu/wtd/.

[6] L. Aalto, N. Gothlin, J. Korhonen, and T. Ojala. Bluetooth and wap push based location-aware mobile advertising system. In *ACM MobiSys*, 2004.

[7] L. Adamic and B. Huberman. Zipf's law and the internet. *Glottometrics*, 3:143–150, 2002.

[8] A. Bakre and B. R. Badrinath. Handoff and system support for indiret tcp/ip. In *The 2nd USENIX Symposium on Mobile and Location-Independent Computing*, 1995.

[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In *IEEE Infocom*, 2006.

[10] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular internet access using unplanned 802.11 networks. In *ACM MobiCom*, 2006.

[11] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *IEEE Infocom*, 2006.

[12] E. Chlebus and R. Ohri. Estimating parameters of the pareto distribution by means of zipfapos;s law: application to internet research. In *IEEE Globecom*, 2005.

[13] Y. Dalal and R. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21:1040–1048, December 1978.

[14] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. Technical report, IETF RFC 2460, December 1998.

[15] D. Goodman, J. Borras, N. Mandayam, and R. Yates. Infostations: A new system model for data and messaging services. In *IEEE VTC*, 1997.

[16] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *IEEE Infocom*, 2001.

[17] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. In *IFIP Networking*, 2005.

[18] D. Ho and S. Valaee. Mobile hot spot in railway systems. In *22nd Biennial Symposium on Communications*, 2004.

[19] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *ACM SIGCOMM Workshop on DTN*, 2005.

[20] Z. Jiang and L. Kleinrock. An adaptive network prefetch scheme. *IEEE Journal on Selected Areas in Communications*, 16:358–368, April 1998.

[21] A. Lindgren and A. Doria. Probabilistic routing protocol for intermittently connected networks. Technical report, draft-lindgren-dtnrg-prophet-01.txt, IETF Internet draft, July 2005.

[22] D. A. Maltz and P. Bhagwat. Msocks: An architecture for transport layer mobility. In *IEEE Infocom*, pages 1037–1045, March 1998.

[23] A. Matsumoto, M. Kozuka, K. Fujikawa, and Y. Okabe. Tcp multi-home options. Technical report, draft-arifumi-tcp-mh-00.txt, IETF Internet draft, October 2003.

[24] A. Nandan, S. Das, G. Pau, M. Gerla, and M.Y.Sanadidi. Co-operative downloading in vehicular ad-hoc wireless networks. In *IEEE Second Annual Conference on WIreless On-demand Network Systems and Services(WONS)*, 2005.

[25] A. Nandan, S. Das, B. Zhou, G. Pau, and M. Gerla. Adtorrent: Digital billboards for vehicular networks. In *IEEE/ACM International Workshop on Vehicle-to-Vehicle Communications*, 2005.

[26] J. Ott and D. Kutscher. Bundling the web: Http over dtn. In *WNEPT*, 2006.

[27] V. N. Padmanabhan and J. C. Mogul. Using predictive prefetching to improve world wide web latency. *ACM SIGCOMM Computer Communication Review*, 26(3):22–36, July 1996.

[28] C. Perkins. Ip mobility support for ipv4. Technical report, IETF RFC 3344, August 2002.

[29] M. Schlager, B. Rathke, S. Bodenstein, and A. Wolisz. Advocating a remote socket architecture for internet access using wireless lans. *Journal of Mobile Networks and Applications*, 6:23–42, 2001.

[30] T. Small and Z. J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *ACM MobiHoc*, 2003.

[31] A. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *ACM MobiCom*, 2000.

[32] T. Spangler. Push servers review. *PC Magazine*, pages 156–180, June 1997.

[33] R. Stewart, Q. Xie, K. Morneault, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream control transmission protocol. Technical report, IETF RFC 2960, October 2000.

[34] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, 2000.