PPWeb: A Peer-to-Peer Approach for Web Surfing On the Go

Ling-Jyh Chen[1], Ting-Kai Huang[1], and Guang Yang[2]

[1]Institute of Information Science, Academia Sinica
{cclljj, tkhuang}@iis.sinica.edu.tw
[2]Nokia Research Center, Palo Alto
guang.g.yang@nokia.com

Abstract— **With wireless technologies extending to every part of our daily lives, mobile networking applications are becoming increasingly popular for accessing the Internet. Among them, web surfing is one of the most important applications because the World Wide Web has accelerated the dissemination of information and knowledge via the Internet unencumbered by geographic boundaries. In this paper, we propose a peer-to-peer approach, called PPWeb, for mobile web surfing. 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 web surfing service than traditional schemes, and thus facilitates more effective web surfing on the go.**

I. INTRODUCTION

The last few years have seen an impressive growth in Internet applications. One striking success in this area has been the World Wide Web (WWW), which has 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 web surfing on the go.

Proper handling of mobility is the key to the success of mobile networking applications. Although recent studies have reported that intermittent network connection is inevitable for mobile users on a daily basis [9] [10]. Grossglauser et al. [15] have 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 surf the web on the go [3] [5] [6] [7] [13] [16] [18] [19] [20] [21] [23]. The approaches can be categorized into three types: offline-based, cache-based, and Infostation-based. Offline-based and cache-based approaches facilitate mobile web surfing by prefetching web contents to a local storage. Infostation-based approaches, on the other hand, support on-demand HTTP requests by deploying dedicated servers as bridges between the Internet and mobile networks. However, the capability of these approaches for mobile web surfing 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 PPWeb, for mobile web surfing applications. In addition to combining the strengths of cache-based approaches (i.e., prefetching the most likely requested web pages to a local storage) and Infostation-based approaches (i.e., allowing on-demand HTTP requests), the PPWeb 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 II, we review related works on web surfing in mobile networks. In Section III, we describe the PPWeb scheme and the *Collaborative Forwarding* feature used to disseminate web content in mobile networks. Section IV presents a comprehensive set of simulation results, which we analyze and explain in detail. We then present our conclusions in Section V.

II. RELATED WORKS

Mobile HTTP-based web access has been researched for a number of years, and several approaches have been proposed to enable web surfing on the go [3] [5] [6] [7] [13] [16] [18] [19] [20] [21] [23]. The approaches can be roughly classified into three types: offline-based, cache-based, and Infostation-based approaches.

Basically, offline-based approaches download web pages to a local storage in advance and allow users to access the documents, even when they are mobile or disconnected. Many widely used tools, such as `wget` [3] and `wwwoffle` [5], fall into this category, and most well known web browsers have built in functionalities to support offline mode operations.

However, the main drawback of these approaches is that mobile users must download web documents manually, and they can only access a limited number of web pages that have been pre-fetched to a local storage.

Unlike offline-based approaches, cache-based approaches automatically download those web pages that are considered likely to be requested in the near future. This is done in either a *push*-based [6] [13] [23] or a *pull*-based [18] [19] [21] fashion. More precisely, when *push*-based approaches are used, the content provider automatically supplies a mobile user with popularly requested web content 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) web content (using its own content selection algorithm) without the HTTP requests issued manually by mobile users. Since cache-based approaches prepare web contents before they are physically requested, they generally perform more efficiently (in terms of service time) than offline-based approaches. The trade-off is that they incur a tremendous storage overhead in return for the performance gain (i.e., the more web pages they cache in the local storage, the greater the likelihood that they will be able to serve the next HTTP request without consuming extra Internet bandwidth). However, this is considered infeasible for emerging power/storage-constrained handheld devices. In addition, offline-based and cache-based approaches only allow Internet-capable users to download web pages, but they do not provide a way for Internet-incapable users to access web content.

Infostation-based approaches provide web access for mobile users by installing Infostations to act as bridges between the Internet and mobile networks [14], 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 web content from Infostations by using local wireless connections (e.g., WiFi and Bluetooth). Hence, mobile web surfing is possible, even when mobile users are not directly connected to the Internet. For example, [16] proposes a solution called *Mobile Hotspots*, which provides mobile Internet access in railway systems. Another approach called *Thedu* [7] uses Internet proxies to collect and pre-fetch web search results from pre-defined search engines; then, mobile users can perform web searches on the local storage of the nearest Infostations. Additionally, in [20], Ott et al. propose deploying Bundle Routers (BR) as Infostations to separate the Internet and DTNs. BRs are responsible for fetching web content from the Internet and forwarding the data to the inner challenged mobile networks (i.e., communication opportunities are intermittent and the network is frequently partitioned). Unlike previous schemes, the BR-based scheme requires mobile users' collaboration to forward data in a multihop and store-carry-and-forward fashion. It also allows users to access web content, even if they cannot locate an Infostation. However, the major shortcoming of the above approaches is that they require dedicated Infostations, which act as gateways to the Internet; thus, they suffer from scalability and single-point-of-failure problems.
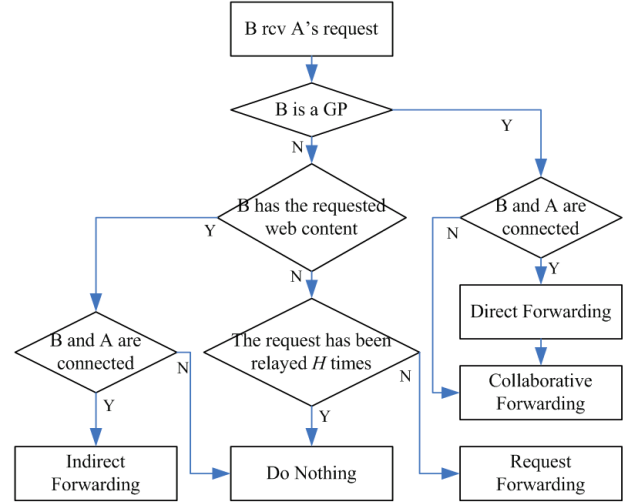

Fig. 1. Illustration of the request process algorithm of the PPWeb scheme.

III. THE PROPOSED APPROACH: PPWEB

In this section, we present our peer-to-peer approach, called PPWeb, for mobile web surfing applications. The approach combines the strengths of offline-based, 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 web surfing services. We describe PPWeb in detail in the following sub-sections.

A. PPWeb Architecture

There are two types of participating peers in the PPWeb 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 PPWeb system, there are two cases when a peer A requests a web page: if A is a GP, he can download the web page 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 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 k increases.

The proposed PPWeb 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 web page from the Internet, and forwards the content

to A if they are directly connected (i.e., by the *Direct Forwarding* algorithm). Next, B disseminates the web content 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 web pages previously requested by peers in the network. This allows the PPWeb system to reduce redundant downloading when multiple peers request the same web pages. Of course, proper buffer management is required to further improve the performance of the PPWeb 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 web content has been cached, and then implements one of the following two options:

a) *If B has the web content requested by A*, he forwards it to A if they are directly connected (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 web page (which depends on the underlying *Collaborative Forwarding* algorithm); whereas the complete web page is forwarded in the *Direct Forwarding* phase.

b) *If B does not have the web content 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 times (i.e., the *Request Forwarding* algorithm); otherwise, he does nothing.

Additionally, the PPWeb 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 HTTP request-reply session, and thus shorten the service time of that transmitted page. Similarly, *Request Forwarding* propagates HTTP requests to the network, which increases the probability that the requests will reach GPs and thus be cached in the network.

B. Collaborative Forwarding Algorithm

In this study, we incorporate the newly proposed HEC-PF algorithm [11] to provide collaborative forwarding for the PPWeb system. The HEC-PF scheme is based on the H-EC scheme [12], which tries to forward the second copy of erasure coded blocks in sequence in H-EC's *Aggressive Forwarding* phase until the end of the network contact period. In contrast, the HEC-PF scheme implements the *Probabilistic Forwarding* algorithm and only enters the *Aggressive Forwarding* phase if a newly encountered node has a higher likelihood of successfully

forwarding the message (called the *delivery probability*) to the destination peer than the current peer.

More specifically, if there are K nodes in the network, we denote the i-th node as X_i, the j-th node as X_j, the accumulated contact volume between the node pair X_i and X_j in the last T time units as t_{X_i,X_j}, and the *delivery probability* for the node pair X_i and X_j with a distance of at most k-hops as $P^k_{X_i,X_j}$. The one-hop delivery probability from the source node (X_S) to the destination node (X_D) is given by the ratio of the accumulated contact volume over the overall contact volume[2], as shown in Eq. 1.

$$P^1_{X_S,X_D} = \frac{t_{X_S,X_D}}{\sum_{i=1}^{K} t_{X_S,X_i}} \quad (1)$$

The two-hop delivery probability, $P^2_{X_S,X_D}$, can be derived by Eq. 2. The equation is comprised of three components: the scaling constant, $\omega_2 \in [0...1]$, which decides the impact of two-hop message transfer on the overall delivery probability; the likelihood value, $1 - P^1_{X_S,X_D}$, which is the probability that a message can *not* be transmitted directly from node X_S to node X_D (i.e., it is impossible to complete the message delivery in one hop); and the sum of the two-hop *transitive* delivery probability based on the *transitive* property, i.e., if node X_S frequently encounters node X_i, and node X_i frequently encounters node X_D, then X_i is a good candidate relay node for forwarding messages from X_S to X_D. Finally, the delivery probability of transferring a message from node X_S to node X_D is given by summing the two delivery probabilities, as shown by Eq. 3

$$P^2_{X_S,X_D} = \omega_2(1 - P^1_{X_S,X_D}) \sum_{\substack{1 \le i \le K \\ i \ne S, i \ne D}} (P^1_{X_S,X_i} P^1_{X_i,X_D}) \quad (2)$$

$$P_{X_S,X_D} = P^1_{X_S,X_D} + P^2_{X_S,X_D} \quad (3)$$

Note that the decision about the *delivery probability* considers the *contact frequency* in the history and the *contact volume*, which represents the proportion of time that the two nodes were in contact in the previous time window. Consequently, the HEC-PF scheme is better able to achieve effective data forwarding in challenged networks.

IV. EVALUATION

We now evaluate the performance, in terms of the service ratio and traffic overhead, of the PPWeb scheme and the Mobile Hotspots scheme using a Java-based simulator, called DTNSIM [2]. 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 web pages 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 web surfers. To be

---

[1]This is the well-known "two-hop" scenario as used in [12] [15] [24].

[2]If $i == j$, $t_{X_i,X_j} = 0$.

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

	γ	PPWeb	Mobile Hotspots	Normalized Overhead
iMote	20%	4,298	1,423	3.02
	40%	3,986	1,591	2.50
	60%	3,638	1,694	2.15
UCSD	20%	211,604	65,723	3.22
	40%	189,702	64,858	2.92
	60%	163,883	62,663	2.62

realistic, the HTTP requests are based on the distribution of the top 500 requested web pages according to the hit-count statistics of the campus proxy server of National Taiwan University for the period Apr. 2006 to Sept. 2006. Each web document is converted into MHTML format [22], which is a MIME (Multipurpose Internet Mail Extensions [8]) HTML document containing one or more objects, such as text, images, or videos.

For simplicity, we assume that web surfers only issue HTTP 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 HTTP request can be relayed at most 2-hops (i.e., H=2) with the replication factor f equal to 4. For the *Collaborative Forwarding* scheme (i.e., HEC-PF), the replication factor of the erasure coding r is set to 2; each coded message is fragmented into 16 equal-sized blocks (i.e., $N = 16$); the sliding time window T (used to estimate the delivery probability) is set to 1,000 seconds; the scaling constant ω_i is set to 0.25; and the buffer size of each mobile peer is unlimited. All the simulation results presented in this section are based on the average performance of 200 simulation runs.

A. Evaluation Scenarios

We evaluate two network scenarios based on realistic campus wireless network traces, namely, the iMote [1] and UCSD [4] traces, which are publicly available for research purposes. Table I 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 [10] [12] [17], 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.

B. Evaluation I: Service Ratio

Here, we evaluate the service ratio performance of the PPWeb 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.

From Figures 2 and 3, we observe that the proposed PPWeb scheme outperforms the Mobile Hotspots scheme 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.

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, PPWeb is superior to Mobile Hotspots in all test cases.

C. Evaluation II: Traffic Overhead

Next, we evaluate the traffic overhead of PPWeb 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 II shows the results where the normalized overhead is derived by taking the ratio of the traffic overhead of the PPWeb scheme over the Mobile Hotspots scheme.

The results in Table II show that the traffic overhead consumed by the PPWeb scheme is about three times more than that required by the Mobile Hotspots scheme, and the normalized overhead decreases as the γ value increases. Note
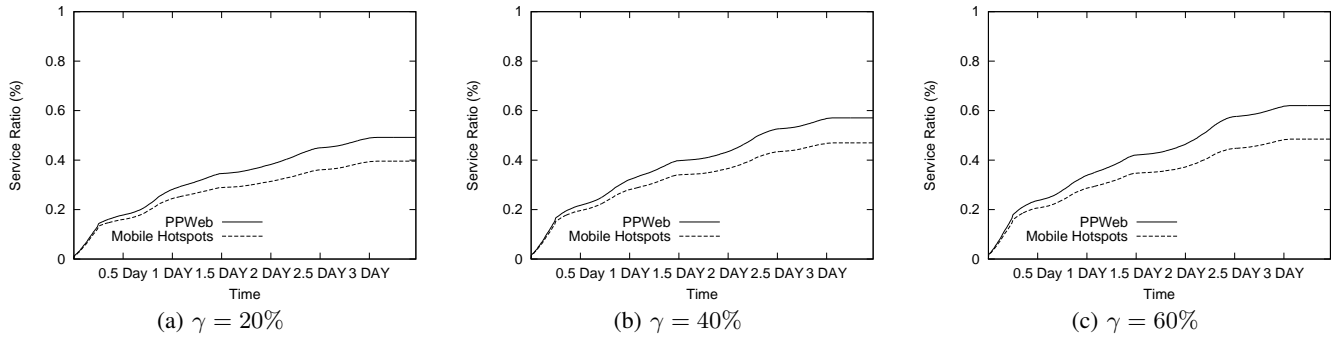
Fig. 2. Comparison of the service ratio performance of the PPWeb and Mobile Hotspots schemes with various numbers of GPs in the iMote scenario ($\gamma = 20, 40, 60\%$)
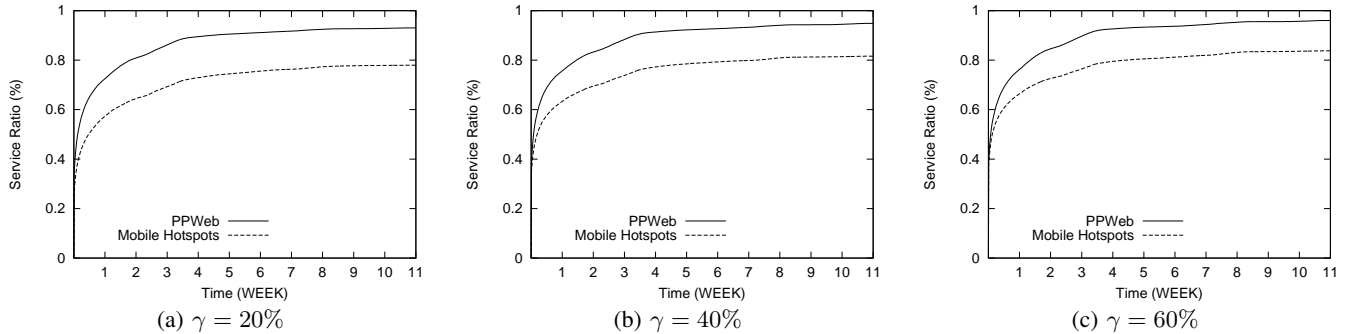

Fig. 3. Comparison of the service ratio performance of the PPWeb and Mobile Hotspots schemes with various numbers of GPs in the UCSD scenario ($\gamma = 20, 40, 60\%$)

that the traffic overhead is upper bound by the collaborative forwarding algorithm. For instance, in our simulation, the upper bound of the normalized overhead is 4 (i.e., 2 due to the replication factor of erasure coding, times 2 due to the aggressive forwarding phase of the HEC-PF algorithm). Of course, the traffic overhead of the PPWeb scheme can be adjusted by tuning the replication factor of erasure coding or replacing the collaborative forwarding algorithm. Generally, the more replicated data stored in the network, the better the service ratio achieved by the PPWeb 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.

V. CONCLUSION

In this study, we have proposed an approach called PPWeb to improve mobile web surfing applications. Unlike traditional approaches, PPWeb 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 PPWeb 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] Crawdad project. http://crawdad.cs.dartmouth.edu/.
[2] Delay tolerant network simulator. http://www.dtnrg.org/code/dtnsim.tgz.
[3] Gnu wget. http://www.gnu.org/software/wget/.
[4] Ucsd wireless topology discovery project. http://sysnet.ucsd.edu/wtd/.
[5] Wwwoffle. http://www.gedanken.demon.co.uk/wwwoffle/.
[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] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani. Web search from a bus. In *ACM MobiCom Workshop on Challenged Networks*, 2007.
[8] N. Borenstein and N. Freed. Mime (multipurpose internet mail extensions) part one: Mechanisms for specifying and describing the format of internet message bodies. Technical report, IETF RFC 1521, Sept. 1993.
[9] 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.
[10] 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.
[11] L.-J. Chen, C.-L. Tseng, and C.-F. Chou. On using probabilistic forwarding to improve hec-based data forwarding for opportunistic networks. Technical Report IIS-07-009, Institute of Information Science, Academia Sinica, 2007.
[12] L.-J. Chen, C.-H. Yu, T. Sun, Y.-C. Chen, and H. hua Chu. A hybrid routing approach for opportunistic networks. In *ACM SIGCOMM Workshop on Challenged Networks*, 2006.
[13] E. Costa-Montenegro, F. Gonzalez-Castano, J. Garcia-Reinoso, J. Vales-Alonso, and L. Anido-Rifon. Pushing web pages into personal digital assistants: Need, tools and solutions. In *Mobile HCI*, 2002.

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

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

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

[17] 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.

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

[19] Z. Jiang and L. Kleinrock. Web prefetching in a mobile environment. *IEEE Personal Communications*, 5(5):25–34, Oct. 1998.

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

[21] 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.

[22] J. Palme, A. Hopmann, and N. Shelness. Mime encapsulation of aggregate documents, such as html (mhtml). Technical report, IETF RFC 2557, March 1999.

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

[24] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure coding based routing for opportunistic networks. In *ACM SIGCOMM Workshop on Delay Tolerant Networks*, 2005.