Using Cooperative Multiple Paths to Reduce File Download Latency in Cellular Data Networks

Danyu Zhu, Matt Mutka and Zhiwei Cen
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
Email: {zhudanyu, mutka, cenzhiwe}@cse.msu.edu
Telephone: (517)-353-4633

Abstract

Current cellular data networks provide relatively low throughput, which results in long delays when downloading large files. We propose a cooperative parallel file downloading scheme to reduce latency in cellular data networks. Mobile nodes form a mobile ad hoc network (MANET) via their IEEE 802.11-based links to share their cellular link capacity. A file is split into pieces and downloaded simultaneously by the client and several neighbor nodes that act as relay proxies. The proxies then use the MANET to forward packets to the client. Thus, the client uses multiple paths for parallel downloading. No special hardware or modifications on the base stations are needed. An on-demand proxy discovery algorithm is proposed for fast and efficient proxy discovery. Experimental and simulation results show that cooperative parallel file downloading may significantly improve cellular network utilization and file download performance.

Keywords: Parallel Downloading, Mobile Ad Hoc Networks, Cellular Data Networks

Using Cooperative Multiple Paths to Reduce File Download Latency in Cellular Data Networks

Danyu Zhu, Matt Mutka and Zhiwei Cen
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
Email: {zhudanyu, mutka, cenzhiwe}@cse.msu.edu

***Abstract*— Current cellular data networks provide relatively low throughput, which results in long delays when downloading large files. We propose a cooperative parallel file downloading scheme to reduce latency in cellular data networks. Mobile nodes form a mobile ad hoc network (MANET) via their IEEE 802.11-based links to share their cellular link capacity. A file is split into pieces and downloaded simultaneously by the client and several neighbor nodes that act as relay proxies. The proxies then use the MANET to forward packets to the client. Thus, the client uses multiple paths for parallel downloading. No special hardware or modifications on the base stations are needed. An on-demand proxy discovery algorithm is proposed for fast and efficient proxy discovery. Experimental and simulation results show that cooperative parallel file downloading may significantly improve cellular network utilization and file download performance.**

***Index Terms*— Parallel Downloading, Mobile Ad Hoc Networks, Cellular Data Networks**

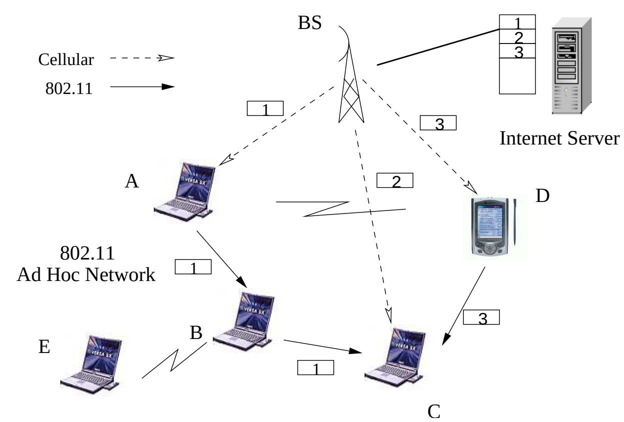


Fig. 1. Cooperative Parallel File Downloading

I. Introduction

With the deployment of new 3G techniques (UMTS, CDMA2000) in cellular data networks and the widespread use of IEEE 802.11-based wireless local-area networks (WLANs), we envision mobile devices that have the ability to access both wireless networks simultaneously. The new GSM/GPRS-enabled iPaq H6315 developed by HP and T-Mobile combines mobile phone technology with Wi-Fi and Bluetooth wireless networking.

Although an IEEE 802.11-based WLAN provides much higher bandwidth in comparison to a cellular network,[1] its small coverage area (up to 250m) makes it impossible for an "always on" Internet connection. When moving outside the range of WLANs, mobile users experience significant performance degradation with the bandwidth limitation on cellular data networks. Significant delay is expected when downloading a large file, such as a MP3 music file from an Internet server via the slow cellular link. On the other hand, bursty Internet usage patterns result as idle cellular links most of the time.

We present a novel cooperative parallel file downloading scheme, which integrates the cellular data network and the IEEE 802.11-based mobile ad hoc network (MANET) to reduce file download latency of the cellular data networks. The basic idea is that the mobile nodes cooperate and share their idle cellular links via a MANET formed from their IEEE 802.11-based interfaces. A file is split into small portions. Several neighbor nodes act as proxies to share the burden of file downloading with the destination node by downloading portions of the file from the Internet server via their idle cellular links and forwarding them to the destination via the IEEE 802.11-based MANET. Thus, the nearby idle cellular links and the high speed IEEE 802.11-based MANET may be utilized to create multiple paths from the Internet server to the destination. Multiple portions of the file are downloaded in parallel via different paths to the destination node, which can significantly reduce file download latency. No special hardware is required in the mobile nodes. The base station of the cellular network is not involved in the file downloading. A suite of protocols are devised to support cooperative parallel file downloading for proxy discovery, ad-hoc routing, and failure recovery.

Figure 1 shows an example of cooperative parallel file downloading, in which five mobile users form a MANET. A and D act as *proxies* to contribute their idle cellular links to support *client* C's file downloading from the Internet server. B is a relay node (*forwarder*) in the MANET, contributing to the system by forwarding packets from the proxy A to the client C. The file is split into small portions and downloaded in parallel via three different paths.

In this paper, we assume that the participating mobile de-

[1] The IEEE 802.11b standard supports up to 11 Mbps and the 802.11a/g standard supports up to 54 Mbps, while current GPRS network supports a maximum bandwidth of 384 Kbps per user. Even the future 3G cellular network, such as CDMA2000 1xEv-DO provides only 38.6 Kbps to 2 Mbps.

vices are equipped with both cellular and IEEE 802.11-based wireless network interfaces. There is no radio interference between the IEEE 802.11-based MANET and the cellular data network since they operate under different frequency band ranges. [2] The Internet file servers support partial file downloading. The HTTP 1.1 byte-range header may be used to indicate which portion of the file to obtain from the web server.

The rest of the paper is organized as follows. Related work regarding the hybrid network and parallel downloading techniques are presented in section II. In section III, the process of proxy discovery and file downloading are introduced. Performance evaluations and results are presented in section IV.

II. Related Work

Parallel downloading has been used as a technique to improve the performance of large web downloads. Rodriguez and Biersack study a dynamic parallel-access scheme to accelerate web downloads by connecting to multiple mirror servers and downloading different parts of the file from each of them [8]. Recent development of peer-to-peer networks also adopt parallel download techniques to accelerate file download from different users sharing the same document [2]. Some special erasure codes, such as Tornado Codes, can be used in parallel downloading [1]. The goal of the cooperative parallel downloading scheme is independent of the parallel-access algorithm used. The techniques and algorithms developed in these research activities can be used to enhance cooperative parallel file downloading.

The idea of utilizing the idle cellular links with the help of IEEE 802.11-based MANETs is directly inspired by the *Cooperating ad Hoc network to sUpport Messaging* (CHUM) project, which has been presented as an approach to integrate 3G networks and ad hoc networks in a manner that significantly reduces 3G network costs to provide support for *instant messaging (IM)* [9], to download multimedia data, such as on-air TV programs [4] and to provide better QoS support by aggregation idle cellular links [11]. A distributed trust model and credit system is proposed in [10] to promote cooperation among mobile nodes. This system can also be applied to the cooperative parallel file downloading scheme described in this paper.

There has been some work in the area of integrating ad hoc and cellular data networks [3]. In [5], mobile users form an UCAN architecture using their 3G and IEEE 802.11-based ad hoc links. A relay proxy helps to forward packets from the base station to the clients with poor channel quality via ad hoc links to improve throughput. UCAN is designed specifically for the 1xEV-DO (HDR) 3G cellular network, which limits its application. It can only utilize one proxy for a client with poor link quality. The maximum bandwidth is limited by the cellular link capacity. The iCAR [7] places ad hoc relay stations at strategic locations to relay signals between mobile hosts and base stations. Bursty traffic could be diverted from one congested cell to another one in order to circumvent congestion. iCAR requires special kinds of relay stations instead of utilizing the existing relay ability provided by IEEE 802.11-based network interfaces. Although it is useful for diverting bursty traffic to nearby idle cells, iCAR does not provide a way to improve utilization of the idle cellular link under light traffic load.

[2]The IEEE 802.11b/g operate on the 2.4-2.483GHz band range and the 802.11a operates on 5.15-5.25GHz, 5.25-5.35GHz and 5.725-5.825GHz band ranges. The 2.5G of cellular networks (e.g., GPRS, EDGE) operate on 800-900MHz (cellular band) and 1.5-1.8GHz (PCS band). Future 3G cellular networks will operate on 1.885-2.023GHz and 2.210-2.200GHz band ranges.

III. Proxy Discovery and File Download

We present an on-demand proxy discovery algorithm for the client to discover possible proxies for a file downloading session. A new round of proxy discovery is needed for each file portion. The additional delay introduced by proxy rediscovery is reduced by pipelining the file transmission request. An extended AODV [6] routing protocol is used to establish the hop-by-hop routing table from the proxy to the client.

A. On-demand Proxy Discovery

The client floods a *proxy discovery request* (PDREQ) within a given range in the MANET to find the potential proxies for a file portion. The PDREQ message carries the client's address (SR), a sequence number (SEQ) that is incremented every new round of proxy discovery, a request broadcast range (RBR) value that is decremented every time the message is rebroadcasted, and the size of the requested file portion.

When receiving a PDREQ, the node compares the SEQ with the largest one it has for the source and drops the PDREQ with equal or smaller SEQ. The node then makes a decision whether or not to act as a proxy or a forwarder, which is based on its current network traffic load, the size of the request portion and other considerations, such as battery power. For a positive RBR and a positive forwarder decision, the node decrements the RBR value and rebroadcasts the request with its address attached in the PDREQ as a forwarder (FR). With a positive proxy decision, the node returns a *proxy discovery reply* (PDPLY). Otherwise, the request is dropped. A node can act as a proxy and a forwarder at the same time. The PDPLY is sent back through the reverse route by the new proxy, containing the SEQ, a sequence of forwarder addresses and the address of the proxy (PR). A timeout value is associated with the PDPLY indicating the time period that the proxy reserves its cellular link for the request, during which a file portion download request must be received from the client.

A set of available proxies is kept in the client for each file downloading session. Newly discovered proxies are inserted into the set. The client uses a pre-defined *degree of parallelism* (DOP), which is the number of proxies plus one (the client's cellular link), to limit the number of proxies used simultaneously in order to reduce collisions in the MANET. The clients selects an available proxy from the set of discovered proxies when (1) downloading a new file portion after finishing downloading one; (2) when receiving a new proxy reply while not exceeding the DOP limit; or (3) when it has a broken connection to a proxy due to mobility. With an empty set, the client starts downloading a file portion via its own cellular link and sends a new PDREQ message for a new round of proxy discovery. Proxy selection is based on hop count, and the round trip time (RTT) estimated via the proxy discovery process.

In Figure 2, six mobile devices form a MANET. Client A broadcasts a PDREQ to its neighbors with SEQ (1), file portion size (1000), broadcasting range RBR (3) and source address (A). C decides to be a proxy and returns a PDPLY. B would like to act as a forwarder but not as a proxy. It attaches its address into the original request and rebroadcasts the message to its neighbor E. With a positive proxy decision, E returns a PDPLY to B, which in turn sends it back to A. Therefore, proxies E and C are discovered by the client A via a round of PDREQ/PDPLY message exchange, agreeing to download a file portion of size 1 Kbyte.

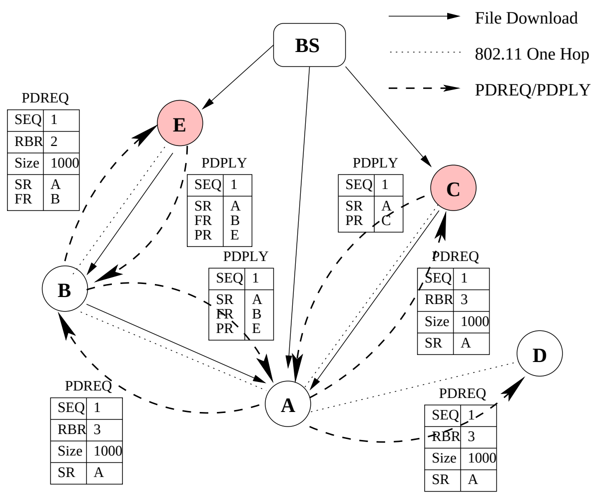


Fig. 2. On-demand Proxy Discovery Example

An extended AODV protocol is used to establish a hop-by-hop routing table from the proxy to the client. The PDREQ and PDPLY message is used in the same manner as the route request (RREQ) and the route reply (RREP) in AODV. When receiving a PDREQ message, a routing entry is inserted into the routing table for the client node. When receiving a PDPLY message, a routing entry is inserted for the proxy node. Therefore, the routing table is established along with the proxy discovery process. The forwarder list associated with the PDREQ message and the PDPLY message is only used by the client to explicitly know who is contributing to the file download.

The on-demand proxy discovery protocol provides a simple and efficient solution for the routing problem. The client is able to locate as many proxies as possible for every file portion even under an unreliable and highly dynamic environment. The discovered proxy can be used for downloading any file portion with the same size as contained in the PDREQ message. Therefore one round of PDREQ/PDPLY message exchange can find several available proxies and initiate several portions to download simultaneously. In the previous example, two proxies are found by one PDREQ message, which can start downloading two portions to the client. On the other hand, the client starts file downloading immediately from its own cellular link, and gradually increases the degree of parallel downloading with the newly discovered proxies. It guarantees that the client has at least the same performance of a normal 3G network download even if it fails to discover any proxies.

B. Pipelining Requests

After selecting a proxy, the client starts downloading by sending a *file portion download request* (FDREQ) to the proxy. It includes the location of the file, the beginning and the ending position of the requested file portion, and the source address of the client. When receiving the FDREQ, the proxy sets up a TCP connection to the Internet file server via its cellular link on behalf of the client and starts downloading the requested portion. The proxy forwards all packets to the client by establishing another TCP connection between the proxy and the client via the IEEE 802.11-based MANET. After finishing downloading of a file portion, the client selects another proxy from the available proxy set or performs another round of proxy discovery with an empty available set. After the client receives all file portions, it reconstructs the document.

For every file portion, there is a preceeding proxy discovery and a connection setup process. To avoid the idle time period during which no data is transmitted (see Figure 3), the proxy discovery and file downloading process may be pipelined. Since it is very likely that a proxy will agree to download another file portion, the client may send a new proxy discovery request directly to a current proxy just before it finishes downloading the file portion. The proxy and forwarders along the route may also make decisions for the new request while the current download is occurring. When the client receives a positive reply from this proxy, a new file download request may be sent without waiting for the current one to be finished. With pipelining, the client may reuse the TCP connection to the proxy to avoid slow start phases and avoid the idle time period of proxy rediscovery and connection setup.

Each file portion should be small enough to provide fine granularity of striping and deal with the unreliability of the proxy. However, it should also be sufficiently large to limit the idle times between block requests in comparison to the transmission time of a block (see Figure 3). On the other hand, pipelining requires a minimum portion size and therefore a maximum number of portions. Suppose S is the file size, and B is the number of portions, the portion size should be such that $\frac{S}{B} > RTT \times \mu$, where μ is the download rate from the client via this proxy and RTT is the round trip time from the client to the Internet server via this proxy.

C. Failure Recovery

The mobile devices can move out of range from the established downloading path, which results in broken routes. To detect and recover from route failure, a timeout that is two times the round trip value from the client to the file server is set up in the client for each proxy in use. If the next packet from a proxy does not arrive to the client by the time the timeout expires, the proxy is considered to have failed. The rest of the file portion will be combined into other portions for the next round of downloading. If the client receives packets from

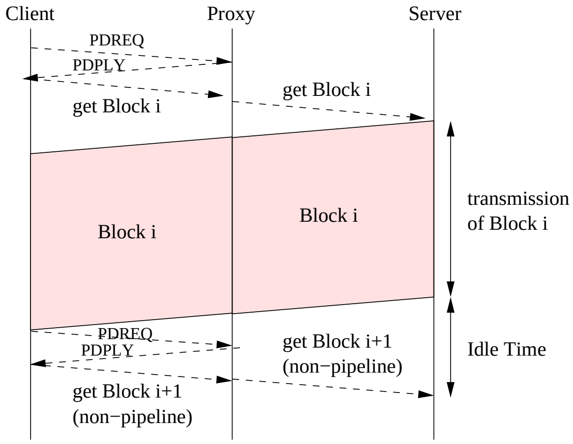


Fig. 3. Block Request without Pipeline

TABLE I
DOWNLOAD LATENCY AND THROUGHPUT GAIN WITH/WITHOUT PARALLEL DOWNLOADING

Client	Proxy1	Proxy2	Latency	Gain
256K	—	—	65.70s	—
256K	256K	—	35.22s	87%
256K	256K	256K	24.51s	168%
512K	—	—	33.02s	—
512K	512K	—	17.60s	87%
512K	512K	512K	12.22s	169%
512K	256K	256K	18.83s	75%
1M	—	—	16.43s	—
1M	1M	—	9.16s	79%
1M	1M	1M	6.66s	147%
1M	512K	512K	10.12s	62%

failed proxies, it will inform the proxies to stop downloading by sending them *stop download requests* (STREQ). The IEEE 802.11 MAC layer calls a call-back function to inform the mobile device of next-hop failure. This feature can be used by the proxy or by the forwarders to inform the proxy to stop downloading when a route failure is detected.

IV. PERFORMANCE EVALUATION

We established a testbed and conducted experiments on a real network environment to justify our motivation in Section IV-A. We also evaluate the performance of cooperative parallel file downloading by various simulation scenarios. The simulation models, metrics, and methodology are shown in Section IV-B. We start with the simplest scenario of a single client, downloading a fixed length file, and then move to scenarios with multiple file downloads in Section IV-C.

A. Experiments

The experimental testbed consists of three Linux-based laptops (C, $P1$ and $P2$), an IEEE 802.11 router/access point, and a Linux server. We use the IEEE 802.11 wireless LAN to simulate the 3G cellular network with the access point acting as the base station. Each laptop is equipped with two IEEE 802.11b wireless cards. One of them operates under infrastructure mode, which connects to the access point via channel 1. The other operates under ad-hoc mode, which connects to the other laptops via channel 11. Since channel 1 and channel 11 use non-overlapping frequencies, the traffic will not have interference between the IEEE 802.11b WLAN and the ad hoc wireless network. The Linux server connects to the router/access point via Ethernet as an Internet file server. C is a client to download a file from the Linux server, and $P1$ and $P2$ are two proxies for C. The downlink bandwidth of the laptops' WLAN interfaces are limited by Linux *iproute2* traffic control tools in order to emulate different downlink bandwidths in cellular data networks. We conduct experiments over different combinations of downlink speed in laptops C, $P1$ and $P2$.

A file server is established in the Linux server that can accept requests to download portions of a specific file. A TCP server is set up in each proxy to accept proxy requests from the client. File downloading requests are sent by the client to a proxy, which in turn sets up another TCP connection to the file server to download the requested part of the file. Packets are forwarded in the proxy between the two TCP connections.

We conducted experiments in three cases to download a 2 Mbyte file: without a proxy, with one proxy, and with two proxies. In the first case, the file is directly downloaded by C via its WLAN interface. For the other cases, the file is split into portions with 20 Kbytes each and downloaded in parallel by C's WLAN interface and the ad hoc interface. When downloading via the ad hoc interface, the data streams are routed from the access point either to the proxy $P1$ or the proxy $P2$ in the WLAN, which further forwards to the client via their ad hoc interfaces. For each downlink bandwidth combination, we conducted the experiment three times and computed the average downloading latency. The downlink bandwidth of C, $P1$, $P2$ and the corresponding downloading latency are recorded in table I. The gain field shows the throughput gain in comparison to the case with no proxy.

The results of the experiments show that the mobile devices can fully utilize the nearby idle cellular links by using cooperative parallel file downloading. The client can achieve up to 169% throughput gain by utilizing two proxies. Because of increasing collisions on the ad hoc network, the gain decreases slightly as the downlink speed increases.

B. Simulation Model

We implement the cooperative parallel file downloading scheme in the $ns2$ network simulator. N mobile nodes are randomly placed in the area of $886m \times 886m$. Each node has a cellular interface to communicate with the base station and the Internet and an IEEE 802.11b interface, where 11 Mbps data rate is supported at the 115 meter range. The cellular interface uses an one-hop routing scheme from the base station to the mobile node. Different cellular interfaces use different radio channels and do not interfere with each other. The base station is located in the center of the simulation area and the mobile nodes can connect to the base station from any location in the simulation area. Node mobility is set according to the random waypoint model.

We use three metrics to evaluate the performance of the cooperative parallel downloading scheme: the file download latency; the minimum, maximum and average performance gain in comparison to a non-proxy download; and the average number of proxies used for each file download.

C. Simulation Results

We start with a simple scenario of a single client with one file to download. We assume that all mobile nodes have the same cellular link bandwidth and limit the proxy request hops to be 3. We vary the size of the download portion from 5K to 950K, the maximum node moving speed from 0m/s, 5m/s to 15m/s, the cellular link bandwidth from 256 Kbps, 512 Kbps, 1Mbps to 2 Mbps, the file size from 100K, 500K, 1M, 2M to 4M, and simulate different client densities by placing 35, 50, 80 and 100 nodes in the simulation area.

Figure 4(a) shows the impact of the size of the file portion under different node density. The results indicate that a small file portion will increase the downloading latency due to the high overhead of routing and limited pipelining. A larger portion does not provide enough modularity for parallel downloading, and also results in performance degradation. It is obvious that the optimum size for this simulation scenario is between 5K to 50K. As expected, mobile nodes more easily find proxies when there is a higher node density.

Figure 4(b) shows the performance gain with different degree of parallelism settings under different cellular downlink bandwidth. The results indicate that the average number of proxies used increases linearly with the increasing of degree of parallelism without affection by the cellular downlink bandwidth. However, increasing the degree of parallelism using a slower cellular downlink results in higher performance improvement. The reason is the IEEE 802.11-based MANET more easily saturates with higher cellular link capacity with the same number of proxies, which is also justified by the experiments. A higher degree of parallelism may be chosen by mobile nodes with a slower cellular downlink capacity. Figure 4(c) reveals the relation between different downloading file sizes and the performance gain. The results show that the larger files benefit more from parallel downloading.

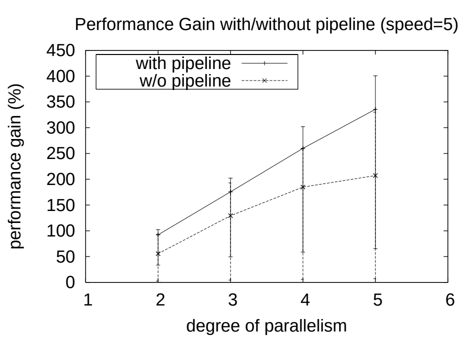


Fig. 5. Effect of Pipeline Technique

Figure 5 shows the maximum, minimum and average performance gain for downloading a 4M file with or without pipelining proxy discovery and downloading requests. Pipelining may significantly increase the system performance. For example, with a degree of parallelism of 5, the average performance increases 127% with pipelining.

The next scenario we investigate has a cellular downlink that is a normally distributed random variable with average 600 Kbps and variance 200 Kbps and limited by a minimum speed of 38.6 Kbps and maximum of 2 Mbps to simulate the real network. We select the node density to be 50, and randomly select 5 or 10 mobile nodes to start downloading files of size 1 Mbytes simultaneously to simulate low and high traffic load. To make the case simple, we assume mobile nodes are self-interested. A mobile nodes will accept proxy requests from other mobile nodes when its cellular link is idle and it always accepts forwarder requests. In order to reduce the contention in the IEEE 802.11 network, one mobile node can only accept one proxy request at a time.

Figure 6 shows the performance gain with 5 simultaneous file downloads under different degree of parallelism and different mobility. The error bars represent the maximum and minimum performance gain. The result shows that the performance increases as we increase the degree of parallelism. With higher node mobility, the IEEE 802.11 connection from the proxy to the client is more likely to break, which slightly decreases the performance gain. Figure 7 shows the similar result with 10 simultaneous file downloads. With higher traffic load, increasing the degree of parallelism has no effect on increasing the system performance. In both figures, the maximum performance gain increases linearly with DOP, which indicates some nodes can fully utilize the parallelism. The minimum performance gain is always positive, which means the performance is at least same as the non-proxy scheme.

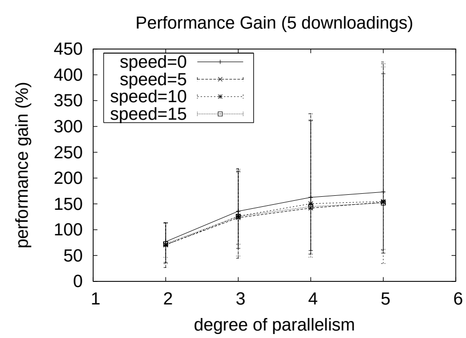
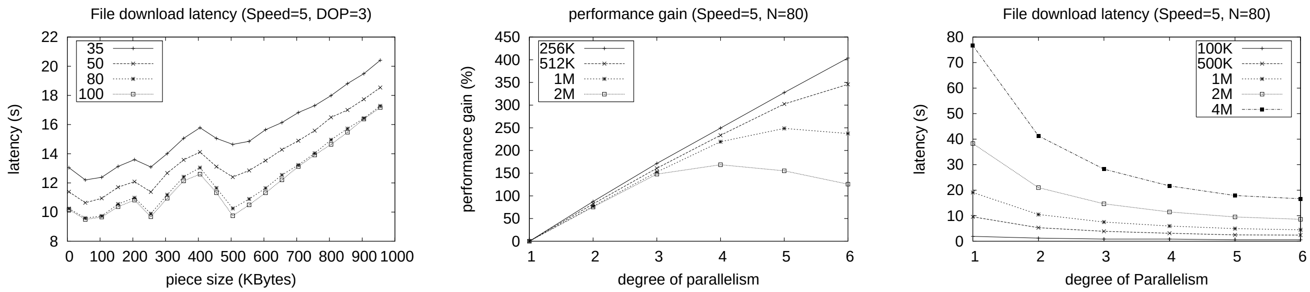


Fig. 6. Performance Gain by 5 Downloads

V. Conclusion

With the popularity of IEEE 802.11-based wireless networks and cellular data networks, we propose a novel cooperative parallel file downloading scheme to utilize multiple paths in the cellular data network with the help of the IEEE 802.11-based mobile ad hoc network. A simple and efficient proxy discovery protocol is used to find new proxies. The experimental and simulation results show that cooperative parallel file downloading can significantly reduce file downloading latency without changing the network architecture.



(a) Node Density (b) Cellular Link Bandwidth (c) File Size

Fig. 4. Performance with different DOP, Density, Cellular Link Bandwidth and File Size

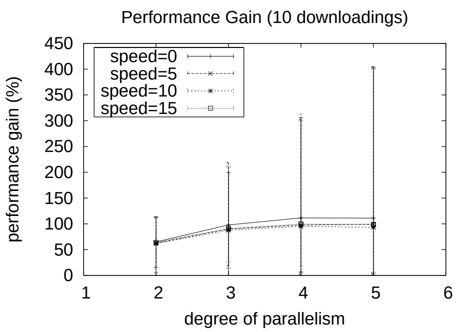


Fig. 7. Performance Gain by 10 Downloads

References

[1] J. W. Byers, M. Luby, and M. Mitzenmacher. Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads. In *INFOCOM*, pages 275–283, New York, NY, Mar. 1999. IEEE.

[2] B. Cohen. Incentives build robustness in bittorrent. In *First Workshop on the Economics of Peer-to-Peer Systems*, June 2003.

[3] H.-Y. Hsieh and R. Sivakumar. On Using the Ad-hoc Network Model in Cellular Packet Data Networks. In *Proceedings of MOBIHOC*, June 2002.

[4] S. Kang and M. Mutka. Efficent Mobile Access to Internet Data via a Wireless Peer-to-Peer Network. In *Proceedings of PERCOM2004*, pages 197–205, March 2004.

[5] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *MOBICOM*, pages 353–367, September 2003.

[6] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proceedings of WMCSA99*, pages 90–100, February 1999.

[7] C. Qiao and H. Wu. icar: An intelligent cellular and ad-hoc relay system. In *Proceedings of IEEE IC3N*, October 2000.

[8] P. Rodriguez and E. W. Biersack. Dynamic parallel access to replicated content in the Internet. *IEEE/ACM Transactions on Networking*, 10(4), Aug. 2002.

[9] D. Zhu and M. Mutka. Sharing Presence Information and Message Notification in an Ad Hoc Network. In *Proceedings of PERCOM2003*, pages 351–358, March 2003.

[10] D. Zhu and M. Mutka. Promoting Cooperation among Strangers to Access Internet Service from an Ad Hoc Network. In *Proceedings of PERCOM2004*, March 2004.

[11] D. Zhu, M. Mutka, and Z. Cen. QoS Aware Wireless Bandwidth Aggregation (QAWBA) by Integrating Cellular and Ad-hoc networks. In *Proceeding of QShine2004*, October 2004.