RC22468 (W0205-216) May 30, 2002
Computer Science

IBM Research Report

CAPS: A Peer Data Sharing System for Load Mitigation in Cellular Data Networks

Young-Bae Ko, Kang-Won Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Thyagarajan Nandagopal
University of Illinois

CAPS: A Peer Data Sharing System for Load Mitigation in Cellular Data Networks

Young-Bae Ko† Kang-Won Lee† Thyagarajan Nandagopal‡

† IBM T. J. Watson Research Center ‡ University of Illinois
Email: {*youngko, kangwon*}*@us.ibm.com* *thyagu@crhc.uiuc.edu*

Abstract

The exponential growth of mobile data users and services places a heavy burden on the limited wireless bandwidth of cellular data networks. The situation will be exacerbated with the advent of high bandwidth multimedia applications for mobile devices. In this paper, we propose an architecture called the *Cellular-based Ad hoc Peer Data Sharing system (CAPS)* to reduce the load on the cellular network while improving request response times. In CAPS, mobile hosts in a cellular network form an overlay multi-hop wireless network. This ad hoc network acts as a 'virtual cache' that enables data sharing among mobile hosts. Participating mobiles share the contents of their local caches with other mobiles. A subset of mobile hosts keep track of the location of objects with minimal overhead. Using CAPS, popular objects can be obtained over the ad hoc network without accessing the cellular infrastructure, thereby reducing the load on the cellular network and improving data access latency. We have extensively evaluated the performance of this architecture through simulations in *ns-2* across a wide range of scenarios, and find that CAPS reduces the load at the base station by up to 60%, and also improves the request response times for users by 40% on the average.

1 Introduction

The popularity and growth of wireless data has led to the development of various systems to enable wireless data on mobile devices, ranging from wide-area cellular data networks, e.g. GPRS [1] and 3G/4G networks [2], to decentralized ad hoc networks, e.g., Bluetooth [3], and wireless LANs based on 802.11 [4] and HIPERLAN [5] standards. The average data rates supported by these systems also vary widely, from 100 Kbps for GPRS networks to up to 54 Mbps for 802.11 based networks. The transmission range typically limits the throughput, hence, the data rates in wide-area cellular networks differ from that of short range data networks, such as 802.11, by a couple of orders of magnitude. This diversity in the types of wireless networks is a fundamental characteristic of the future because no single system is best suited for all environments. Consequently, mobile users are likely to access data using multiple wireless networks, for example, using an 802.11/HIPERLAN network at home and at work, and using a 3G/4G cellular data network while on the move.

In both current and emerging wireless architectures, access to the Internet backbone is controlled either by some form of a cellular network or by wireless local area networks (WLANs) called 'WLAN hotspots.' Since it is not possible to have ubiquitous coverage through WLAN hotspots, mobile users requiring continuous connectivity will have to be able to connect through the cellular data network. Backbone access is required when the desired data are not in the local cache or if they are real-time in nature. With the exponential growth in wireless data users and the projected linear-scale growth in cellular data bandwidth, the demand on the resources of the cellular channel will be very high, and the data access latency of the system may increase without bound. This is because, even if the base station fetches the requested objects from the backbone network with negligible delay, these objects may get queued at the last hop due to the limited cellular bandwidth. This situation will only be exacerbated with the use of multimedia applications on mobile devices.

In this paper, we propose a novel solution to alleviate the wide-area wireless bandwidth scarcity problem by integrating ad hoc networks within a cellular data network architecture. The key idea behind the proposed scheme, that we call the *Cellular-based Ad hoc Peer Data Sharing system* (CAPS), is to establish an overlay multi-hop wireless network among mobile hosts on a separate channel and then try to locate and retrieve popular objects from the local cache of peer mobiles, using a distributed directory. When the requested object cannot be obtained from a peer mobile, the request is sent over the backbone via the cellular channel. Thus, peer mobiles participating in CAPS are collectively viewed as a 'virtual proxy cache' or a 'file sharing system.' The key contributions of CAPS are as follows:

- *An overlay architecture that transfers a portion of the load from the cellular network to an ad hoc network formed by hosts in the cellular data network.*
- *A simple algorithm and protocol for efficient data access among peer hosts in the ad hoc network, based on object location information maintained at a few directory hosts.*

The CAPS system consists of simple low overhead protocols for establishing the ad hoc peer network for data sharing via automatic configuration among mobile hosts. We expect that CAPS will reduce the load on the cellular system, thereby reducing the object download time, or response time, observed by wireless users. Therefore, we can think of a situation wherein the cellular service provider favors and supports CAPS as an additional load mitigation scheme under heavy load conditions. However, cellular network support is not a necessary condition for the deployment of CAPS since it can operate in a fully distributed manner. We present the operation of CAPS in both scenarios.

From mobile users' perspective, there exists an economic motivation to look for their data amongst peers. Given the limited wireless bandwidth available to service providers and the broadcast nature of the wireless channel, end users are charged for every byte of data transferred beyond a certain threshold. With the expected growth in wireless data and applications catering to mobile devices, users may be reluctant to pay for data that can be easily found from the cache of peer mobile hosts.

The rest of the paper is organized as follows. Section 2 describes related work in this area, and Section 3 presents the network and service model considered in this paper. Section 4 provides an overview of the proposed scheme, and Section 5 details the operation of the proposed CAPS architecture and the directory algorithm. Section 6 quantifies the performance of CAPS through simulations. We conclude the paper with a discussion of future work in Section 7.

2 Related Work

There exists a wide body of related work aiming to utilize ad hoc networks to augment some facet of cellular systems, such as range, routing, throughput, availability, etc. Many of these studies focus on using ad hoc networks to improve the connectivity and range of the cellular network, and to provide data access to disconnected hosts. In this section, we present brief descriptions for a representative sample of these studies.

The Multi-hop Cellular Network (MCN) system [6] effectively extends the reachability of cellular networks by transmitting data over multi-hop paths consisting of mobile devices. In iCAR [7], special ad-hoc relay stations (ARSs) are deployed to relay traffic from a congested cell through a neighboring uncongested cell. A similar relaying scheme has also been proposed in [8], wherein the use of dual mode mobile stations having both GSM and ad hoc interfaces is introduced to improve coverage and capacity of the present GSM cellular network. While CAPS shares the common idea of deploying ad hoc networks within a cellular network, the role of the ad hoc network is different. In CAPS, the ad hoc network is used to alleviate the bandwidth scarcity problem, whereas MCN and iCAR try to provide an extended reachability and efficient routing, respectively.

In [9], the authors propose a data sharing system for mobile devices, called 7DS, and show how popular data can be disseminated over a set of mobile devices disconnected from or partially connected to a cellular network. Their model assumes a push/pull operation to exchange data between directly reachable mobile hosts. It also performs routing and relaying functions for disconnected hosts. Even though 7DS uses the notion of *information sharing* just like our work, the end goal is different. We focus on load mitigation and throughput enhancement to the cellular network, while 7DS aims to provide extended data availability when the infrastructural support is not available. Moreover, 7DS focuses on the effectiveness of distributing a single data object to the mobile host upon contact within a single hop wireless connectivity. In this paper, we consider mobile users accessing multiple objects in a multi-hop ad hoc network.

In [10], the authors have studied the trade-off between cellular networks and ad hoc networks and showed that peer-to-peer communication over ad hoc networks has benefits in terms of throughput, delay, and power consumption. Our study poses a related but different question on the effectiveness of utilizing ad hoc networks in parallel with a cellular infrastructure.

On a different note, cellular service providers are investigating techniques to provide seamless roaming between Wi-Fi networks and cellular data networks to provide seamless computing. The Wireless Internet Service Provider Roaming (WISPr), a subcommittee of the Wireless Ethernet Compatibility Alliance (WECA) [11], has been investigating this issue. The focus is on establishing an unified billing and location tracking standard across these heterogenous networks. To the best of our knowledge, there has been no prior work relating to the use of ad hoc networks to supplement data services for load mitigation in cellular networks, and this paper proposes some initial solutions to this important problem.

Our scheme is also distantly related to the notion of Community Wireless Networks or Neighborhood Area Networks (NANs), that are used to extend the reach of wired network connections to entire communities [12, 13, 14]. Mobile hosts can bypass the base station and access the backbone using a NAN, if they can participate in one. However, the primary goal of a NAN is to efficiently share the Internet connectivity through wireless LAN, whereas we try to address the relative scarcity of cellular bandwidth. Moreover, these high speed backbone access are not readily available everywhere. The same discussion also applies to the use of wireless 'hotspots' or wireless local area networks that are used to provide Internet services at public places such as airports, restaurants and hotels. Similarly to NANs, they cannot provide ubiquitous coverage, and therefore mobile hosts in a cell that does not have connectivity through NANs or wireless hotspots need alternate load mitigation techniques such as the one proposed in this paper.

Recently, a lot of attention has been drawn to peer-to-peer data sharing systems in the Internet, and new architectures such as CAN [15] and Chord [16] have been proposed for the same. In such systems, the object location information is organized into a distributed hash structure using a sophisticated algorithm so that the search cost is minimized. However, such distributed location management may not be viable in a highly dynamic environment as in mobile ad hoc networks where the group of participants and the network topology can change continuously in a relatively small time scale. Therefore, in this paper, we strive to provide a simpler yet more practical solution to the peer data sharing problem in ad hoc networks.

3 Network and Service Model

In this section, we describe our network and service model. The network model describes an abstract cellular environment which characterizes a large class of (wide-area) cellular networks.

3.1 Network model

Mobile user: Each mobile host has at least two wireless interfaces, of which one interface is used to communicate with the cellular base station, and the other interface is used to communicate with other mobile users. We assume that all mobile users will communicate in the ad hoc mode using a common interface (e.g., 802.11). Each mobile device has limited battery power, and has sufficient storage to accommodate the requirements of CAPS.[1] We do not make any other assumptions on the computing power of the mobile hosts or on the mobility patterns of users as they communicate with each other. Mobile hosts can always reach a base station in the cellular network, though they might not be able to contact other peers on the ad hoc network.

Ad hoc network: We consider a typical ad hoc network of mobile hosts. Links are bidirectional, allowing a wide choice of MAC and routing protocols. We do not place any restrictions on the protocols used to configure and maintain connectivity in the ad hoc network. We simply assume that if a mobile host can reach a peer through the ad hoc network, then there exist suitable protocols to handle connection and session management. For our simulations, we used TCP/IP over the DSDV routing protocol [17] with 802.11 MAC. The maximum data rate of the ad hoc network is typically higher than that of the cellular network, and the ad hoc data range is shorter than that of the cellular network. For example, the peak data rate is up to 11 Mbps for 802.11b and 54 Mbps for 802.11a covering about 250 meters [4], compared to the data rates of 400 Kbps outdoors and 2 Mbps indoors for 3G cellular networks, whose cell radius is typically around 1 km. Note that the effective throughput of a 802.11 based ad hoc network decreases with host mobility and the distance between hosts, due to the need for more error protection. To account for this in our simulations, the ad hoc network has a data rate of 2 Mbps, which is considerably lower than the peak rate of 802.11 systems. The operational frequency of the ad hoc network is separate from that of the cellular network hence there is no interference. For example, GPRS operates in 900 MHz band whereas 802.11 operates in either 2.4 GHz or 5 GHz range depending on the standard.

Cellular data network: We consider a macro cellular network environment with multiple channels per cell. The channel is slotted in time for uplink and downlink communications as in GSM and GPRS. We assume that the cellular network handles only data requests.[2] Mobile hosts access the backbone through a local base station, which handles scheduling. The base station communicates with mobile hosts using a control channel and data channels. The control channel is used by hosts to transmit requests for sending data, and by the base station to announce the schedules. The base station can also use control channel to coordinate CAPS operation such as directory node election if it supports CAPS (see Section 5.1). A cluster of base stations are connected to the Mobile Switching Center (MSC) which serves as the gateway to the rest of the Internet. It also handles call setup and termination, and mobility of hosts by managing hand-offs. In practical systems, the MSC might be responsible for all functions with respect to mobile hosts with the base station merely acting as a transceiver. In this paper, we do not make a distinction between the MSC and the base station, since we do not require these two entities to be distinct.

3.2 Service Model

Mobile hosts run a HTTP client and try to access Web objects from the backbone network. We assume that the objects are spatially correlated (e.g., commuters access news, traffic, local restaurant information, etc.). These objects are assumed to be fresh in the cache up to a few minutes, and are also assumed not to be user-specific. This implies that data with real-time requirements (such as stock quotes) and personal data objects (such as mailbox) have to be served by the base station only.

[1] As we will see later in the paper, object location information can be stored in a few hundred kilobytes.

[2] If the cellular network handles both voice and data, then the description will apply to the data component alone.

4 Architecture Overview

In this section, we present a high-level overview of the CAPS architecture and motivate some of our design choices.

4.1 Extending the cellular architecture with peer data sharing

As mentioned earlier, in our architecture, mobile hosts form ad hoc networks on a separate wireless channel while operating in a cellular network. The resulting ad hoc network can reach mobile hosts across multiple cells as long as connectivity among the mobile hosts is maintained. The data rate of the resulting ad hoc network is typically much greater than that of a cellular network.

The main idea of CAPS is to locate and retrieve popular objects from peer mobile hosts via ad hoc communication whenever possible, instead of retrieving them from the backbone network through the base station. In other words, CAPS treats a group of mobile hosts connected to the ad hoc network as a "virtual proxy cache" or a "data sharing system" that can serve the requests for most popular objects in the cell. If a substantial amount of data requests can be served by ad hoc peer-to-peer (P2P) operation, it will reduce congestion at the base station, and subsequently lower the response time observed by the end users.

The key features of the CAPS architecture are as follows.

- *Easy deployability*: CAPS can be deployed quickly on any cellular network since it requires minimal support from the base station. When the cellular service provider is oblivious of the CAPS protocol, CAPS-enabled mobiles can still operate in a fully distributed manner. We revisit this issue in the next section.

- *Generic in nature*: Our service model includes any data objects that are not freshness-sensitive or user-specific. This encompasses most popular objects (such as MP3 files, video clips, and local information) that will benefit from caching. All application data that comply with this service model can be used with CAPS.

- *Flexibility*: CAPS is not tied to any particular routing, scheduling or MAC protocol used in the ad hoc network of mobile users. In addition, the caching strategies can also be different from host to host.

- *Autonomy*: Individual mobiles control their level of participation in the system, from controlling the number of local cache objects and cache size, to choosing the power level threshold for participation. The only requirement of a mobile host in the ad hoc network is that it should execute the peer-cache agent, and participate in the *directory* election process, to be described below.

4.2 Ad hoc object discovery

When trying to enable peer data sharing over ad hoc networks, we must address the fundamental challenge posed by the peer-to-peer principal, namely "How to determine the location of the target object?" One simple solution to this question could be to search for the target object using a scoped broadcast on the ad hoc network every time a mobile wants to make a request. Whoever has the knowledge of object location can respond to the query. While this approach is easily implementable with no overhead of maintaining object location, it may introduce a large volume of control traffic (e.g., see Gnutella [18]), and therefore may not scale well.

Another way to address this question can be active snooping. Each mobile host actively snoops downlink communications to learn of object delivery information: "which mobile host downloaded which object when." In theory, mobiles can snoop the downlink communication by listening to the *entire* cellular channel. However, active snooping may not work well in practice due to technical and administrative constraints. For example, active snooping is hard to implement in CDMA or FDMA-based cellular networks. While it is possible to snoop the entire channel in a TDMA-based network, we find that energy consumption induced by active snooping is prohibitively expensive for battery-operated mobile devices (see Section 6.2.6). Moreover, due to privacy and security reasons, snooping others' communication may not be allowed.

In CAPS, the object location information is managed by a subset of mobiles called the *directory nodes* (or *dnodes*). CAPS selects only those mobiles, who have enough battery power ($P \geq P_{th}$) to become dnodes because directory nodes must consume extra energy for location discovery and serving location queries. When a non-directory mobile wants to find an object from a peer, it first contacts one of the dnodes to get location information. Depending on the level of support from the cellular service provider, dnodes employ different mechanisms to learn of object location.

First, when the cellular network is oblivious of CAPS, each mobile voluntarily reports its local cache entry to a neighboring dnode, which subsequently updates the location directory. In addition, dnodes passively snoop the ad hoc traffic to discover object location information. This idea is similar to the ones used for route snooping in some ad hoc routing protocols [17].

In case when the cellular network is cooperative, CAPS expects the base station to periodically send down the summary of object transmission during the last interval on the broadcast channel. This information is used to update the location directory of the dnodes. Note that this base station-assisted location update incurs less overhead than the previous case because in this case a single broadcast message updates all dnodes in the cell. Additionally, dnodes can snoop ongoing transmission on the ad hoc channel to update their location directory as in the previous case. In the remainder of this paper, we describe the CAPS architecture with an assumption that the cellular service provider is cooperative.

4.3 Electing directory nodes

The dnode needs to listen to the location information from the base station, snoop traffic on the ad hoc channel, and serve requests for object information in addition to its normal functions as a mobile host. Consequently, dnodes expend more energy than non-dnodes. In CAPS, new dnodes are elected periodically to avoid depleting the battery power of particular hosts. This is done by running a directory election algorithm among hosts.

There were a few general guidelines that we followed when designing the dnode election algorithm.

1. *Control Overhead:* The election algorithm should not induce a large amount of control traffic either on the ad hoc channel or on the cellular channel.

2. *Power Requirements:* The algorithm should consider the battery power constraints of each mobile host as the key metric in electing dnodes. Specifically, we assume that mobile hosts do not participate in the directory election process when their power level drops below a threshold, P_{th}, which is host-specific.

3. *Cellular System Support:* The algorithm must leverage the presence of the cellular infrastructure. In CAPS, the base station can serve as a coordinator for the election algorithm. It signals election intervals, and announces the dnodes for a given interval. A fully distributed directory election scheme can be used, however, if the cellular network is uncooperative.

4. *Accessibility:* To reduce the latency to locate target objects, every mobile host must be able to access a dnode with the lowest possible delay. To ensure this, we can try to identify dnodes in the cell such that every mobile host has at least one dnode in its neighborhood. This is equivalent to that of finding a dominating set[3] in the connectivity graph of the ad hoc network of mobile hosts.[4] Ideally, we would

[3]Given a graph $G = (V, E)$ consisting of a set of vertices V and a set of edges E, a dominating set $S \subset V$ is a set such that every node in V is either in S or is a neighbor of a node in S. A dominating set with minimum cardinality is called a minimum dominating set. Finding a minimum dominating set is an NP-complete problem [19].

[4]The connectivity graph for the ad hoc network is vastly different from the connectivity graph of the cellular network, since the

like dnodes form a minimal dominating set so that the management overhead and power consumption is minimized. However, it may not be always possible to find the minimal dominating set that also satisfies the power constraints.

In the next section, we will detail the peer-to-peer operation of CAPS and the dnode election algorithm.

5 Ad hoc peer-to-peer operation in CAPS

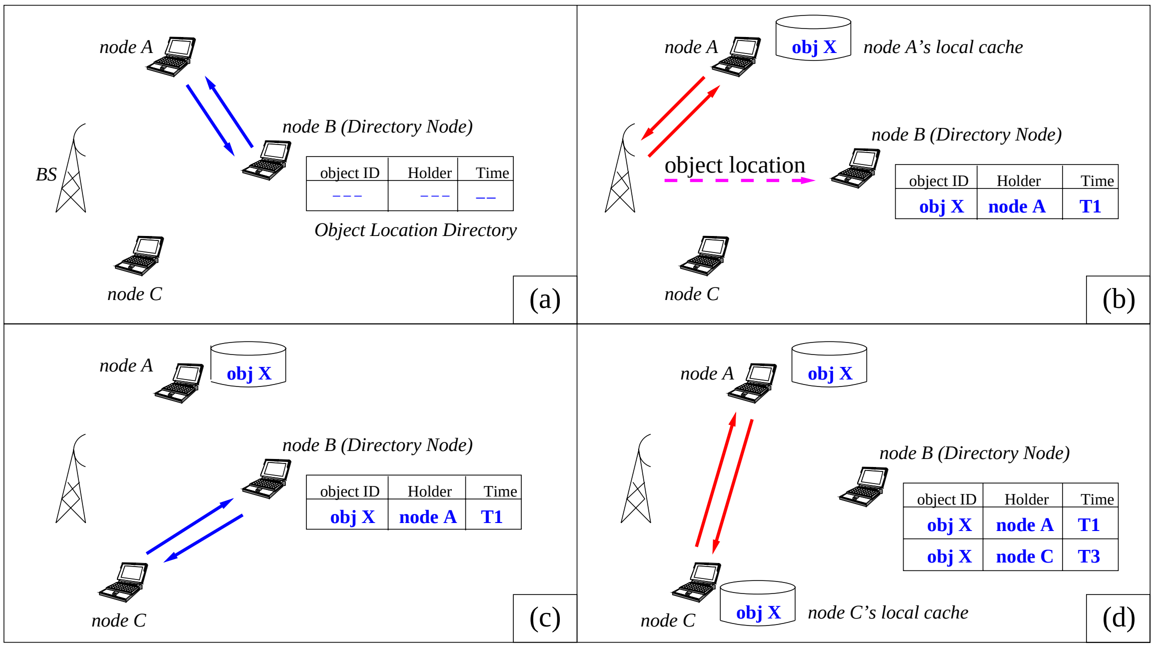


Figure 1: A simple illustration of CAPS operation in a cellular network

Figure 1 illustrates a simple example of the proposed scheme in the extended cellular architecture. There are three mobile hosts A, B, and C in the cellular range of base station BS, where B is a dnode. At time T_0, when mobile host A initially wants to download object x, it first queries the dnode B and discovers that no mobile has cached the object yet (Figure 1(a)). Then mobile host A contacts BS to download the object from the origin site at time T_1. Some time later, BS informs the dnode B with the location of object x. And this information is inserted in B's location directory: (`obj x` $\rightarrow$ `host A` at `time` T_1) (Figure 1(b)). At time T_2, when mobile host C wants to download object x, it first sends a query to the dnode B. Upon reception of the query from C, mobile host B hashes the object ID and finds that mobile A has cached the object. Then it responds to mobile host C with the address of mobile host A and the download time T_1 (Figure 1(c)). Then mobile host C directly contacts A to download object x instead of making a request to the origin server via cellular network at time T_3 (Figure 1(d)). The dnode B updates its location directory that mobile C has also cached object x.

In general, downloading an object at a mobile host involves the following steps.

1. **Decision making**: The mobile host first determines whether it wants to download the target object from the backbone site through the cellular network or to retrieve it from a peer mobile host on the ad

radio range of these two interfaces vary.

hoc network. To reduce cost, mobile users will always try to get the object using the ad hoc peer-to-peer operation whenever possible. This is especially true for large objects such as popular music files or video clips. However, certain types of object such as personal data (e.g. e-mail), real-time data (e.g. stock quotes), and dynamic data (e.g. database queries), must be retrieved from the backbone. Therefore, applications must provide a hint on object types to help the local CAPS agent in decision making process.

2. **Ad hoc peer-to-peer operation**: Once the mobile host has decided to retrieve the object using the P2P communication, it first queries a dnode to obtain the potential location of the object in the cache of some peer mobile host. If the directory node identifies a peer as the location of the object, then the mobile host contacts the target peer for the object. Failure can occur in five cases: (a) when the dnode indicates that the requested object is not in the cache of any other host, (b) the dnode does not respond within a prespecified timeout period, (c) the target peer no longer has a copy of the object in its cache, (d) the route to the target is lost during data transfer, and (e) the target peer is unresponsive. Whenever failure occurs, the mobile host reverts to the cellular mode to obtain the object.

3. **Cellular mode operation**: If a mobile host decides to retrieve the target object from the backbone site, or if the retrieval in ad hoc mode fails, then the mobile host downloads the object in the cellular mode. In this mode, the mobile host makes a request to the backbone server through the base station. The object is returned by the server, or by a cache en route. This is the same as the default mode of operation in wide-area cellular networks.

In step 2, ideally, a mobile host should contact the closest directory node for location query. However, this condition cannot be guaranteed without the knowledge of geographical location information. In our scheme, mobile hosts always try to query dnodes that are within a direct reach. If there are no dnodes in the range, the mobile host randomly selects a dnode. Consequently, if there are enough number of dnodes, then the first scenario (single hop query) will be a common case. For those few nodes that are located outside the range of any dnodes, location queries will be equally distributed among the dnodes because of random selection.

5.1 Directory node operation

As briefly mentioned in the previous section, directory nodes perform three main functions: (1) obtain location information from the base station, (2) serve object location information to requesting peer hosts, and (3) snoop ad hoc communication to update object location information. To maintain a location directory of the objects that are stored by mobile hosts, each dnode maintains a simple hash table: {`object ID` → list of (`mobile address, download time`)}, and responds to a location query with the corresponding (host address, download time) entries. Upon reception for the response from the dnode, the mobile can choose to contact any peer node from the entry. In our scheme, the mobile host first tries to contact a peer node in its wireless range. If it fails, then it tries to contact a peer node that has cached the object most recently. In this section, we describe the dnode election algorithm and several issues related to maintaining the location directory in the ad hoc network.

5.1.1 Electing dnodes

We adapt the distributed constant time minimum dominating set (MDS) approximation algorithm used in [20] for computing the set of dnodes in the ad hoc network.

Our dnode election algorithm uses the following procedure: Periodically, a group of new dnodes are elected in a cell. At the beginning of each period, base station announces the *directory election phase*. Upon this announcement, all hosts with power $P \geq P_{th}$ run the MDS approximation algorithm, described below.

Once this phase is completed, each host that has been elected itself as the dnode reports to the base station. The base station then broadcasts the list of dnodes to all mobile hosts in the cell.

MDS Approximation Algorithm: Let V' be the subset of mobile hosts with power $P \geq P_{th}$ and their immediate neighbors. We want to compute a dominating set among the hosts in the set V', such that all hosts in the dominating set have sufficient power. Given a dominating set S of V', we define the dominator of a host v to be either v, if $v \in S$, or the neighbor of v in the set S. The dominating set acts as the set of dnodes for the next interval.

Consider a mobile host $u \in V'$, with $d(u)$ neighbors denoted by the neighborhood set $N(u)$, dominator $dom(u)$, and effective degree $d^*(u)$, where $d^*(u)$ is the number of its neighbors in V' who have chosen u as their dominator. Let $P_f(u) = 1$ if node u has power $P \geq P_{th}$, and 0 otherwise.

When the base station announces the directory election phase, each host u in V' does the following steps.

1. Host u waits for a random backoff interval to avoid collisions with other ongoing transmissions.

2. u then broadcasts the triplet $[u, d^*(u), d(u), P_f(u)]$ to its neighbors.

3. If u does not have a dominator, then it sets $dom(u) \leftarrow v$, where v is the node in $N(u)$ with the largest value for $[d^*(v), d(v)]$, in lexicographic order, such that $P_f(v) = 1$. Note that u may choose itself as the dominator. In other words, based on the information it has so far, u chooses the highest degree power-eligible node in its neighborhood $N(u)$ with the maximum effective degree.

4. u then sends v a unicast message informing v of its choice. Host v then increments $d^*(v)$.

5. If $d^*(u) > 0$, then u joins the dominating set.

The resulting set is a dominating set in V', though it may not be a minimum dominating set. Once the dnode set is calculated, the base station is informed, which then broadcasts this information to all hosts in the system. Note that if the base station does not coordinate dnode election, the MDS algorithm is initiated by time-outs or topology changes. For the quality of the MDS approximation algorithm, we refer the interested readers to [20].

5.1.2 Object location directory data structure

Each dnode maintains the location information using the MD5 digest values of the URLs [**?**]. In this way, we can store and look up cache entries using the 128 bit digest values instead of variable length URL strings. A similar technique is used by the Cache Digest protocol [21] of Squid server that is designed for exchanging cache content information amongst servers. In the location directory, each hash entry consists of a linked list of the location data structure. Each location data structure consists of three components: (a) the IP address of the mobile host that stores the object (4 bytes), (b) the object download time represented in seconds (2 bytes), and (c) a checksum of URL (2 bytes). The download time is set to 0 when the dnode has been elected and gets incremented every second. With 16 bits, the 'time' field does not overlap within 18 hours, which is a large enough period for our purpose. The URL checksum calculates the 16 bit one's complement of the digest value to resolve potential collisions in the MD5 hash. Using this data structure, 10,000 location data structure can be stored in about 100KB of memory. Note that for a single object there can be more than one location data structure if the object has been downloaded by multiple mobile hosts.

5.1.3 Updating location directory

As described in the previous section, dnodes periodically receive object location information from the base station. In addition, dnodes can snoop the communications on the ad hoc channel to discover up-to-date location information. For instance, if a dnode overhears that mobile host A is downloading object x from mobile host B, then it adds a directory entry specifying that mobile A has cached object x.

In addition, when a dnode receives a location query for object y from mobile host C, then it can infer that mobile host C will cache object y without overhearing the actual transmission. While host C may fail to download the object from the designated peer node for various reasons, it is expected that mobile host C will eventually download object y (in the worst case using the cellular channel). Therefore, it is safe to assume that mobile host C has cached the object y.

It is evident that these partial updates may create inconsistent view among dnodes, since an ad hoc transmission overheard by a dnode A is not necessarily heard by dnode B. Therefore, dnodes need to periodically exchange messages to update each other with local changes. For this purpose, each dnode send out the incremental changes that it made in the last time period T_{update}. In this way, the location information of popular objects, that are cached by a large number of mobile hosts in the cell, gets propagated to all dnodes.

5.1.4 Handing off location directory

To ensure non-disruptive services when a new set of dnodes commences in a new period, the retiring dnodes must transfer their location directory to the new dnodes. To reduce the control traffic during this *hand-off* phase, the predecessor transfers the location directory entries for the most recently accessed objects. In other words, they transfer only the directory entries that has been created within the last $T_{history}$ seconds, where $T_{history}$ is a period during which the location information is considered valid. Ideally, the time period $T_{history}$ must be adaptive to the dynamics of the ad hoc network: if the ad hoc network is highly dynamic then the value of $T_{history}$ must be small since the corresponding mobile host may have left the ad hoc network already. In practice, however, it is hard to determine the dynamics of ad hoc networks. Hence, we employ a conservative value of $T_{history}$, estimated from a long term measurement.

6 Performance Evaluation

In this section, we evaluate the proposed scheme via a simulation study using an extended version of the network simulator, *ns 2 (version 2.1b8)*. We extended the ns simulator to implement the Web access capability on the mobile nodes, ad hoc mode operation among peers including directory query and response, and dnode management scheme. Here, we briefly describe our simulation model, and provide the simulation results under a wide range of simulation parameters.

6.1 Simulation Model

In our simulation, initial locations (X and Y coordinates) of the mobile nodes are set using a uniform distribution. Initially, we test with 10 nodes, moving around in a rectangular region of size 1000 m x 1000 m according to the random waypoint mobility model. In this model, each node chooses a direction, moving speed, and distance of move based on a predefined distribution and then computes its next position *P* and the time instant *T* of reaching that position. The mobility of the random waypoint model is determined by the maximum speed of the mobiles and the pause time after reaching the position. The *maximum* speeds of the nodes were set to 1, 5, 10, 15, 20 m/sec. We ran our simulations with movement patterns generated for different values of pause time, but we present only those with a pause time of 0 seconds, corresponding to continuous motion, as they represent the worst case scenario in terms of mobility.

parameter	*range*	*default value*
arrival rate (req/sec)	0.2 – 0.8	0.4
max speed (m/sec)	0 – 20	5
number of mobile hosts	10 – 40	10
Zipf parameter	0.6 – 1.4	1.0

Table 1: Ranges of the simulation parameters and their default values

The wireless link bandwidth between mobile hosts is set to 2 Mbps to model WLAN speed. Mobile hosts can connect to base station (BS) using a wireless link of 400 Kbps. This rate corresponds to the effective data rate of 3G wireless networks. The wired connection between the base station and the origin server is a duplex link of 10 Mbps with 50 ms delay.

To model the user access pattern, we assume that they will behave in a similar way as the Web users today (see Section 6.2.4 for further discussion). We use ProWGen synthetic trace generator [22] to generate Web access traces, that has been shown to generate realistic traces. ProWGen effectively models key parameters that determine the user behavior, e.g., object popularity, object size distribution, correlation between object size and popularity, and temporal locality. Among these parameters, we focus on the object popularity distribution since they have a direct impact on the caching performance [22, 23]. It has been shown that the popularity of Web objects follows a Zipf-like distribution, i.e., the frequency of access to the i^{th} popular object is proportional to $\frac{1}{i^{\alpha}}$ where α ranges from 0.6 to 1.4 in most Web proxy and server traces. In our traces, the object size distribution follows log-normal distribution with the mean of 7 KB to simulate the Web environment. In Section 6.2.4, we study the sensitivity of the CAPS performance to the object popularity.

6.2 Simulation Results

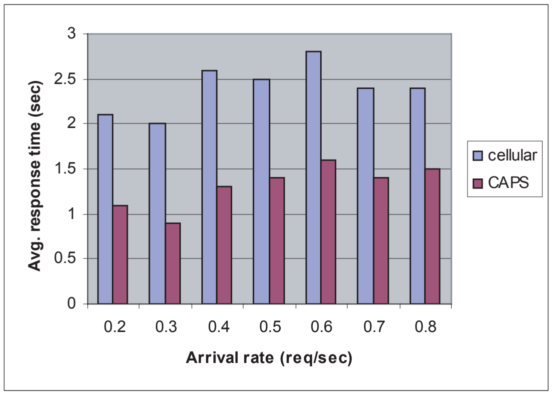
We now present the performance of CAPS in comparison to the basic cellular network under various simulation scenarios. In particular, we study the impact of the following parameters: (a) request arrival rate, (b) host mobility, (c) number of dnodes, (d) number of mobile hosts in the cell, and (e) user access pattern. When varying one parameter, the other parameters are set to their default values. We list the simulation parameters and their default values in Table 1.

For the performance metric, we use the average response time for downloading objects and the number of time-out events. We call a request has been *timed out*, when the desired object could not be downloaded within a certain time value $t_{timeout}$. In our simulation, we set $t_{timeout} = 100$ seconds. Note that a single Web page typically consists of multiple objects.

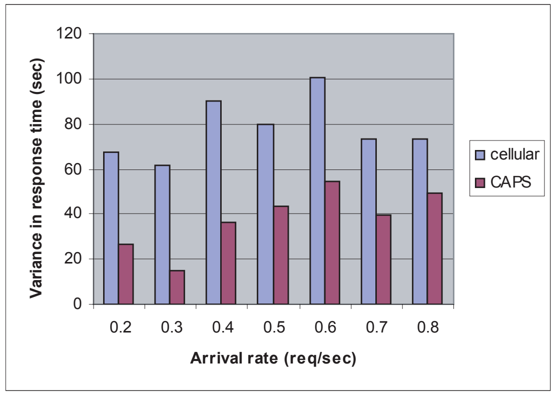
In this paper, we present the case when the mobile hosts are not constrained by the battery power so that every host assumes the dnode functionality and maintains the location information in CAPS. Clearly, this is the best case for CAPS. We compare this with the case when there is no dnode. In Section 6.2.5, we study the impact of having various numbers of dnode and show that a few dnodes can provide good performance. All simulations have been run for 25,000 simulation seconds, which is a long enough period to ensure both schemes to converge to steady states.

6.2.1 User request arrival rate

We first consider how each scheme performs under various load conditions by changing the user request arrival rate. The request arrival rate is defined as the total number of requests made in the cell per second, i.e., the higher the arrival rate the higher the load on the network. Figure 2 presents the average and the variance of response times with respect to the request arrival rate. When calculating the average and variance, we



(a) Average response time



(b) Variance in response time

Figure 2: Impact of the request arrival rate on the latency to download objects

considered only those samples that are not timed out, i.e., whose response time was less than 100 seconds. Recall that we considered 10 mobile hosts moving up to 5 m/sec speed in the cell.

Figure 2(a) shows two general trends. First, the average response time asymptotically increases as the load increases for both basic cellular and CAPS cases. Second, in any load condition CAPS provides shorter response time than the basic cellular network does. In addition to the improved average performance, we observe that CAPS also provides much smaller variance in response time as illustrated in Figure 2(b). Overall, we observe about 30 – 50% reduction in the average response time.

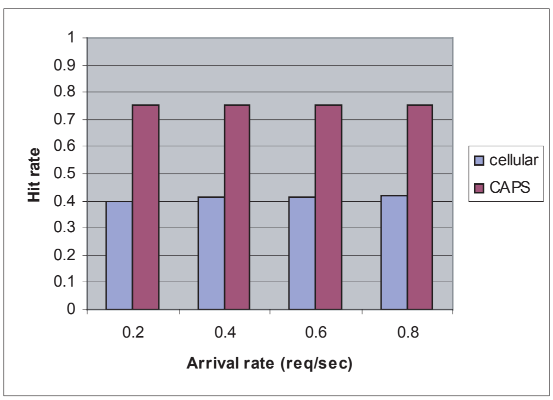


Figure 3: The object hit rate in the basic cellular network and CAPS

The performance improvement of CAPS results from ‘cache hits’ in other mobile hosts’ caches on the ad hoc network. A large fraction of object requests are served by peer mobile hosts over the ad hoc network, reducing the load on the cellular network and avoiding wide-area latency. From the trace log, we find that about 40% of the requests have been served by the local cache in both cases across all load conditions (see Figure 3). In addition to the local hits, about 35% of the requests have been served in ad hoc mode without consuming cellular bandwidth in CAPS. Since mobile hosts always try to download using the ad mode first, these objects are never sent on the cellular network. This means that CAPS effectively reduces the traffic on the cellular network by about 60%.

The advantage of CAPS is more eminent when we consider time-out events. From the mobile user’s perspective, time-out events are equivalent to data unavailability. In our simulation, time-outs happen due to

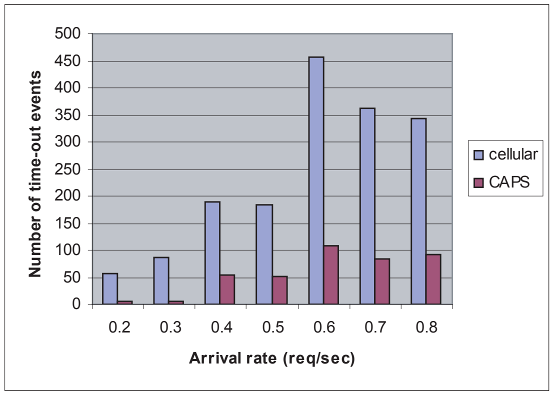
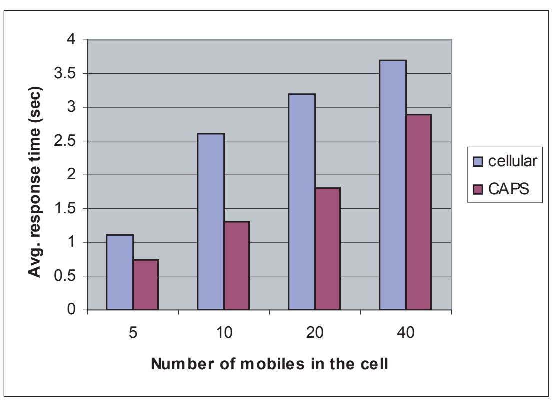


Figure 4: The number of time-out events in the basic cellular network and CAPS

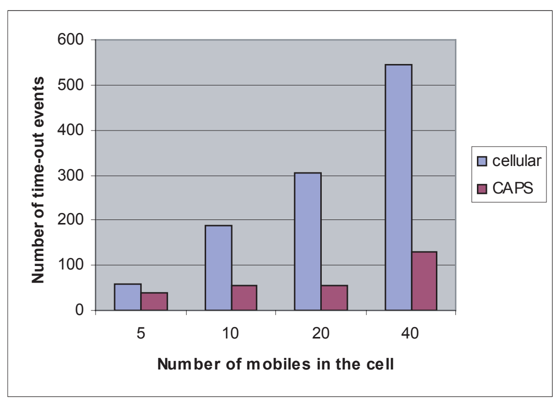
excessively long delay incurred by the congestion at the cellular channel. When the cellular channel becomes overloaded, the packets get dropped at the MAC layer. After the predefined number of unsuccessful retries, the MAC layer gives up retransmission, and eventually the dropped packet must be retransmitted by TCP. Due to the burstiness of user access pattern, some packets experience excessively long delay. Figure 4 plots the number of time-out events in the cellular network and CAPS. With the increase of the arrival rate, the number of time-outs increases for both the cellular system and CAPS. However, the increase of time-outs in the cellular network is much greater than that of CAPS, and therefore, the benefit of CAPS is even greater when the system heavily overloaded.

Overall, we observe that there is a significant performance gain from CAPS under a wide range of load level (40% reduction in average response time and 60% reduction on the cellular traffic) and the performance benefit is more pronounced as the load increases.

6.2.2 Number of mobile hosts



(a) Average response time



(b) Number of time-out events

Figure 5: Impact of the number of mobile hosts in a cell on the latency to download objects

In this section, we study the impact of the population of mobile hosts in a cell on the performance of the cellular network and CAPS. We fix the request arrival rate to be 0.4 and the max speed to be 5 m/sec. Figure 5 presents the results with varying number of mobile hosts in the cell. From the figure, we observe that the average response time and the number of time-out events increase with the number of mobile hosts in the cell. This must be due to increased level of contention on the cellular network and the ad hoc channel. Contentions on the wireless channels result in packet loss and link level retransmission which subsequently affect the end user performance in an adverse manner. In all the cases, however, the effectiveness of CAPS is visible reducing the average response time by 20 – 50% and the occurrence of time-outs by 30 – 80%.

6.2.3 Host mobility

In this section, we consider how host mobility impacts on the performance of the cellular network and CAPS. Using the random waypoint model, we changed the maximum speed of mobile hosts from 1 to 20 m/sec. This corresponds to a wide range of mobility from the walking speed of a pedestrian (3.6 km/h) to the driving speed on a highway (72 km/h). We present the case when the pause time is 0, which corresponds to continuous motion. In this section also, we present the case when there are 10 mobile hosts and the request arrival rate is 0.4 requests per second.

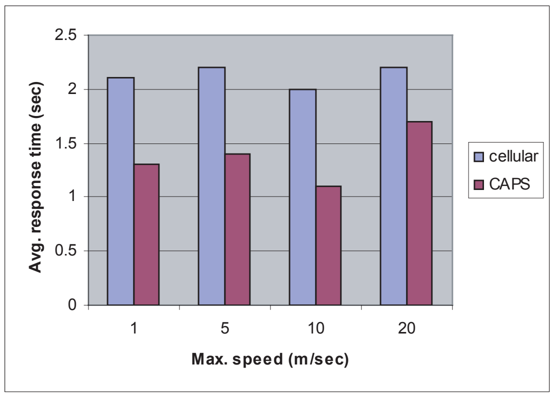


Figure 6: Impact of the host mobility (max. speed) on the latency to download objects

Figure 6 presents the average response time with respect to various mobility scenarios. From the figure, we observe that the average response time of the cellular network is more or less the same across all mobility scenarios, while that of CAPS increases slightly. As the speed of mobile hosts increases the data transfer on the ad hoc network experience higher packet loss due to increased mobility. In general, we observe that the benefit of CAPS is highest when the user mobility is small to medium range. While the benefit of CAPS slightly decreases as host mobility increases, the performance of CAPS is always better (20 – 40% reduction in response time) than the basic cellular network in all cases.

6.2.4 Impact of user access patterns

As previously mentioned, we do not know how the user access behavior will look like in future cellular data networks. However, there are two hints. First, the user access pattern in the wired Web environment has been extensively studied in recent years and the user access patterns (e.g., object popularity distribution, file size distribution, and the correlation between the popularity and size) have been shown to follow certain statistical characteristics [24]. For example, the Web object popularity distribution is highly skewed and heavy-tailed following a Zipf-like distribution. Second, Adya et al. [25] have studied the behavior of PDA and cellphone users browsing Web objects. In that study, the authors have found that the majority of user requests are concentrated on a small number of objects as in the wired case. However, the object popularity distribution does not exactly follow Zipf-like distribution. Instead, the popularity distribution exhibits discontinouity in a few places, and each segment roughly agrees to a Zipf-like distribution curve, whose α parameter ranges from 0.8 to 12.

Based on the above two observations, we have modeled the user access pattern using a synthetic Web trace generator in this paper. In [23], Lee et al. have shown that the benefit of multiple caches cooperating each other is quite sensitive to the user behavior, in particular, object popularity distribution and access density (i.e., the average number of access per object). In this paper, we present the impact of object popularity distribution on the performance of the cellular network and CAPS.

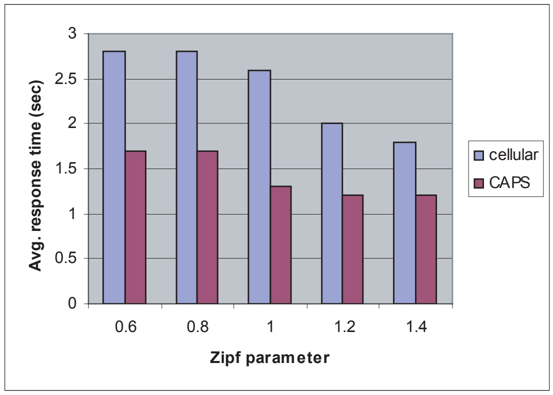


Figure 7: Impact of the object popularity on the latency to download objects

Figure 7 plots the average response time when the Zipf parameter α changes from 0.6 to 1.4. The other parameters were fixed to the default values. When α is large, the user access is highly concentrated on the most popular objects. On the other hand, when the α is small, the user access is less concentrated and is more widely spead to less popular objects. Therefore, caching will be most beneficial when α is high since the most popular objects are likely to be stored in the local cache and later accesses to them will be served locally.

Our results corroborate this arguement showing that the response time of both cellular and CAPS cases decreases as the Zipf parameter increases. The decrease is more noticeable in the cellular case. This is because when α is high the most popular objects can be stored in the local cache of a single mobile host therefore the benefit of single cache becomes more noticeable. This result is consistent with the findings of Lee et al. [23], which reported the benefit of cooperative caching decreases as α increases. However, in all scenarios, CAPS provides better response time and less time-outs than the cellular case (30 – 40% reduction in response time).

6.2.5 Impact of the number of dnodes

So far we have presented the results for the case when all mobiles maintain location information in comparison to the basic cellular network case to show the maximum achievable benefit of CAPS. In this section, we examine the impact of electing a subset of mobiles as dnodes.

Figure 8 presents the average response time when there are 1, 2, and 10 dnodes. The 0 dnode case in the figure corresponds to the basic cellular case. From the figure, we observe that even with a few directory nodes, CAPS provides most of its benefit to the mobile users. While a more detailed analysis on the performance impact of the directory nodes under various operation scenarios is an ongoing research, we expect that a relatively small number of dnodes can provide good performance.

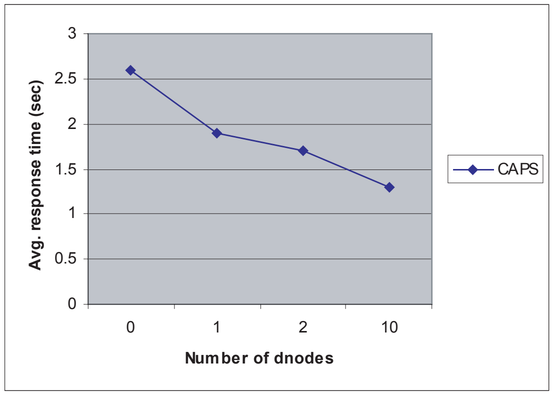


Figure 8: Impact of the number of dnodes on the latency to download objects

	Tx	Rx
Ad hoc	1.6 * S / 2e06	1.2 * S / 2e06
Cellular	1.6 * S / 4e05	1.2 * S / 4e05

Table 2: Energy consumption model where S represents data size in bits

6.2.6 Energy consumption

Battery power is a limited resource on the mobile devices and has to be managed efficiently. In this section, we summarize the implication of peer data sharing on the energy consumption at mobile hosts. We employ the simple energy consumption model implemented in *ns*. In particular, *ns* calcultes the consumed energy at each wireless network interface based on the equations presented in Table 2.

We consider a scenario when there is only one dnode among the total ten mobile hosts in the cell and the dnode doesn't change. This is a worst case scenario for the dnode since all the object location query must be handled by a single dnode. Recall that the base station periodically provides a summary of object download information during the last interval. The request arrival rate is set to 0.4 and the simulation duration 25,000 simulation seconds.

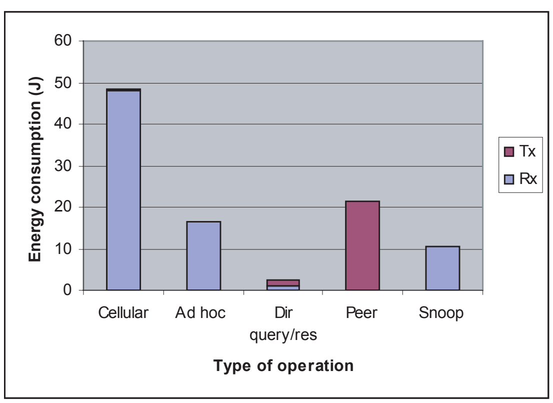


Figure 9: Energy consumption on various types of operation

Figure 9 presents the energy consumption for various types of operations at the dnode. The y-axis represents the consumed energy in J. The x-axis represents each type of operation. The first two columns "Cellular" and "Ad hoc" represent the case of downloading objects *for its own consumption* from the cellular network and the ad hoc network, respectively. We observe that downloading objects from the cellular network is the dominant source of energy consumption expending about 50% of the total energy. The "Dir query/res" column corresponds to the operations to handle object location query and response from the other mobile hosts. Note that as we increase the number of dnodes, the energy consumption in this category will decrease correspondingly. From the figure, we observe that handling directory queries is not a major overhead for the dnode even in this worst case scenario. The "Peer" column represents the case of peer-to-peer data transfer to serve other mobile hosts. Note that non-dnode mobiles will also observe this overhead. Overall, 20% of the total energy consumption is utilized to serve the data requests from the other mobile hosts. The "Snoop" column represents the energy consumed while receiving summary information from the base station. We observe that this extra energy consumption is moderate; about 10% of total energy consumption. However, when there is no support from the base station and if the dnode has to snoop all downlink communication in a TDMA network, the energy consumption is excessively high (120 J per 5 min) prohibiting battery-operated nodes to work as dnodes.

7 Conclusion

The exponential growth of mobile data users and services places a heavy burden on the limited wireless bandwidth of cellular data networks. The situation will be exacerbated with the advent of high bandwidth multimedia applications for mobile devices. On the other hand, the diversity in the wireless data access technology, ranging from 3G/4G wide-area cellular data network to 802.11/HIPERLAN, will likely require mobile users to equip multiple network interface for seamless always-on data access.

In this paper, we proposed an overlay peer data sharing architecture called the *Cellular-based Ad hoc Peer Data Sharing system (CAPS)* to reduce the load on the cellular network while improving request response times experienced by mobile users. In CAPS, mobile hosts in the cellular network form an overlay multi-hop wireless network. This ad hoc network acts as a 'virtual cache' or a 'file sharing system' that enables data sharing among peer mobile hosts. Participating mobiles share the contents of their local storage with other mobiles. A subset of mobile hosts keep track of the location of objects with minimal overhead. Using CAPS, popular objects can be obtained over the ad hoc network without accessing the cellular infrastructure, thereby reducing the load on the cellular network. At the same time, mobile users can enjoy peer data sharing and download without paying for air time to the wireless data service provider.

We have extensively evaluated the performance of this architecture through simulations in *ns-2* across a wide range of scenarios including various load levels, host mobility, population in the cell, and different user access patterns. From the simulation study, we find that CAPS reduces the load on the cellular network by up to 60%, and also improves the request response times for users by 20 – 40% on the average depending on the scenario. We also evaluated the power consumption due to CAPS operation since mobiles typically operate on batteries. We find that about 20% of total energy is consumed for serving other mobiles in peer-to-peer manner and about 10% of total energy is consumed for maintaining object location in the cell.

In this paper, our evaluation primarily focused on the case where the cellular network is cooperative. When the cellular network is not supportive, however, we expect that CAPS can still provide performance enhancement. In such a scenario, the location of the target location has to be queried via scoped broadcast unless the information is already stored in the local cache. The range of the query should be carefully determined (usually limited to a few hops) to avoid long delays in locating objects and reduce control traffic on the ad hoc network. We plan to investigate the CAPS performance in such environment as part of our future work.

References

[1] R. Kalden, I. Meirick, and M. Meyer, "Wireless internet access based on GPRS," *IEEE Personal Communications*, vol. 7, pp. 8–18, Apr. 2000.

[2] "3G Partnership Project." http://www.3gpp.org.

[3] "The Bluetooth wireless info website." http://www.bluetooth.com/.

[4] "IEEE P802.11, The Working Group for Wireless LANs." On-line document, Jan. 2002. `http://grouper.ieee.org/groups/802/11`.

[5] "HiperLAN2 Global Forum." On-line document, Mar. 2002. `http://www.hiperlan2.com/`.

[6] Y.-D. Lin and Y.-C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications," in *Proceedings of IEEE INFOCOM*, Mar. 2000.

[7] C. Qiao and H. Wu, "iCAR: an Intelligent Cellular and Ad-hoc Relay System," in *Proceedings of IC3N*, Oct. 2000.

[8] G. N. Aggelou and R. Tafazolli, "On the relaying capability of next-generation GSM cellular networks," *IEEE Personal Communications*, vol. 8, pp. 40–47, Feb. 2001.

[9] M. Papadopouli and H. Shulzrinne, "Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices," in *Proceedings of ACM MobiHoc*, Oct. 2001.

[10] H.-Y. Hsieh and R. Sivakumar, "Performance Comparison of Cellular and Multi-hop Wireless Networks: A Quantitative Study," in *Proceedings of ACM Sigmetrics*, June 2001.

[11] "Wireless Ethernet Compatibility Alliance." On-line document, May 2002. `http://www.weca.net`.

[12] "The Corner Internet Network vs. the Cellular Giants." New York Times, Mar. 2002. `http://www.nytimes.com/2002/03/04/technology/04MESH.html`.

[13] "OpenNAP project by the Bay Area Wireless Users Group." On-line document, Dec. 2001. `http://www.bawug.org/`.

[14] "SBAY Wireless Project." On-line document, Dec. 2001. `http://www.sbay.org`.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proceedings of ACM SIGCOMM*, Aug. 2001.

[16] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications," in *Proceedings of ACM SIGCOMM*, Aug. 2001.

[17] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, pp. 153–181, 1996.

[18] J. Ritter, "Why Gnutella Can't Scale. No, Really." http://www.darkridge.com/ jpr5/doc/gnutella.html, February 2001.

[19] P. Crescenzi and V. Kann, "A Compendium of NP Optimization Problems." http://www.nada.kth.se/ viggo/wwwcompendium/wwwcompendium.html, March 2000.

[20] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1454–1465, August 1999.

[21] M. Hamilton, A. Rousskov, and D. Wessels, "Cache Digest specification - version 5." http://www.squid-cache.org/CacheDigest/cache-digest-v5.txt.

[22] M. Busari and C. Williamson, "On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics," in *Proceedings of IEEE INFOCOM*, Apr. 2001.

[23] K.-W. Lee, K. Amiri, S. Sahu, and C. Venkatramani, "On the Sensitivity of Cooperative Caching Performance to Workload and Network Characteristics," in *Proceedings of ACM Sigmetrics (extended abstract)*, June 2002.

[24] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *Proceedings of ACM Sigmetrics*, May 1998.

[25] A. Adya, P. Bahl, and L. Qiu, "Analyzing the Browse Patterns of Mobile Clients," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.