iShare: Exploiting Opportunistic Ad hoc Connections for Improving Data Download of Cellular Users

Long Vu[†], Ivica Rimac[‡], Volker Hilt[‡], Markus Hofmann[‡], Klara Nahrstedt[†]

[†] Department of Computer Science, University of Illinois, IL61801, USA

Email: {longvu2,klara}@cs.illinois.edu

[‡] Bell Labs / Alcatel-Lucent, NJ07733, USA

Email: {rimac,volkerh,hofmann}@bell-labs.com

Abstract

This paper presents an Incentive-based Sharing (iShare) protocol that blends cellular and ad hoc networks for content dissemination services. With iShare, mobile users download content from a source via cellular links and at the same time form a mesh ad hoc network for peer-to-peer exchange of content data. The mesh remains robust to network dynamics, minimizes ad hoc communication overhead, and parallelizes the downloading process among mesh members. In order to counter selfish behavior, we apply an efficient and practical "tit-for-tat" incentive mechanism, which exploits proximity and mutual content interest of mobile users. This mechanism becomes particularly effective in the case of network dynamics since we utilize promiscuous and broadcast modes of the ad hoc channel. As a result, our protocol effectively helps to free resources in the cellular network and accelerates the content download for its users. Furthermore, it enables users to continuously obtain data via ad hoc connections during cellular handoff periods and provides multi-homing downloads for groups spanning adjacent cells. We evaluate the performance of iShare by means of simulations and compare it to other content dissemination schemes using cellular broadcast channels, cellular unicast channels, and tree-based protocols. The obtained results show that iShare significantly outperforms alternative approaches and creates a win-win situation by improving performance of both iShare and other mobile users.

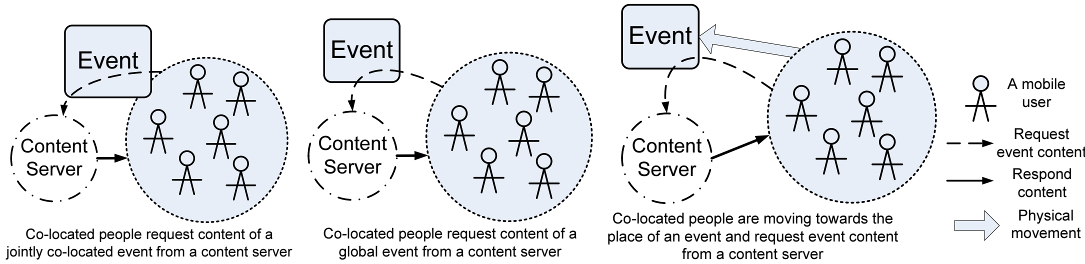
1 Introduction

The proliferation of mobile devices and swift advances of wireless technology lead to the emergence of pervasive computing paradigms. Cellular networks and cellular today allow mobile users access to the Internet and bandwidth-consuming content services including images, music, and video anytime and anywhere. However, cellular access networks can hardly follow the increasing demand of mobile users. This may lead to extended periods of high contention for air link resources during busy hours, thus incurring longer response times and more system failures.

A straight-forward solution to the above problem is an upgrade of the cellular network to increase the capacity, e.g., by deploying more and smaller cells, licensing more spectrum, or even changing the underlying technology. However, these options require changes, deployment and management of cellular infrastructure incurring significant costs. On the other hand, modern cellular devices today already come equipped with additional wireless interfaces such as IEEE 802.11 (wifi) and Bluetooth. These interfaces can be leveraged to use unlicensed spectrum for short range ad hoc communication without any additional investment in network infrastructure. Basically, cellular devices today allow to combine cellular and ad hoc connectivity to leverage the strengths of both technologies. More specifically, base stations in cellular networks have very long transmission ranges (up to 20 km) but relatively limited bandwidth (2.4 Mbps in 3*G* networks). In contrast, the transmission range of a wifi interface is very short (only around 250 m) while it provides high throughput (11 Mbps and 54 Mbps in IEEE 802.11b and IEEE 802.11a/g, respectively).

Some recent research projects have focused on combining cellular and ad hoc connectivity on cellular devices [1, 10, 12, 11, 14, 16]. Most of the published approaches are based on the construction of a tree of ad hoc nodes rooted at high-data-rate proxy nodes connected to the cellular network. The proxy nodes receive data from the source over cellular links and relay them to the final receivers via the ad hoc tree. These protocols usually require additional functionalities being added to the cellular infrastructure such as specific scheduling algorithms and proxy-selection methods. Moreover, the construction and maintenance of a tree structure incurs a high overhead in terms of ad hoc communication, especially under network dynamics.

Besides the aforementioned technical drawbacks of existing approaches, they also lack realistic and applicable incentive mechanism for user collaboration, e.g., by assuming general unselfishness and total cooperativeness among nodes [3, 11, 16, 18]. In more realistic scenario, however, users might not necessarily turn on their wifi interfaces to forward data to other users for free. To address the free-riding problem and create incentives for user cooperation, various techniques such as credit accounting and rewarding have been proposed [4, 7, 15, 19, 21]. However, these mechanisms are complex and applying



(a) Scenario class 1 (b) Scenario class 2 (c) Scenario class 3

Figure 1. Co-located mobile users download similar content at approximately the same time

them requires changes in the business policy of the service providers.

In this paper, we present an Incentive-based Sharing (iShare) protocol that leverages cellular and ad hoc connectivity on mobile devices to provide improved content dissemination services without requiring any changes to or additional functionality from the underlying cellular and ad hoc networking layer. Mobile users download content from a source via cellular links and at the same time leverages ad hoc connectivity to also exchange content fragments following the peer-to-peer paradigm.

The iShare protocol is based on mesh formation using ad hoc connectivity among co-located mobile users, which makes the solution very robust to network dynamics and minimizes ad hoc communication overhead. Furthermore, it parallelizes the downloading process among mesh members, which yields an accelerated download performance for its users. At the same time, fewer air resources are requested from the cellular network, which can be leveraged by the cellular network for data flows effectively improving the performance of other users and applications. Finally, our design includes an efficient and practical incentive mechanism based on "tit-for-tat" reciprocation that helps to enforce cooperation among members and counter selfish behavior. By utilizing promiscuous and broadcast modes of the ad hoc channel, our incentive mechanism is particularly suited and effective in mobile scenarios with network dynamics.

The remainder of the paper is organized as follows. We first present scenarios where the co-located mobile users download similar content of a social event from the content server at approximately the same time, and the system model in Section 2. These scenarios motivate the design of the iShare protocol, which is presented in detail in Section 3. Section 4 evaluates iShare protocol and provides a comparison with alternative schemes based on cellular broadcast/unicast channels and tree-based protocols. We present related work in Section 5 and conclude the paper in Section 6.

2 System Model

In this section, we first present motivation and a classification of scenarios for the design of the iShare protocol. Then, we present system models and design objectives of iShare.

2.1 Scenario Classification

Cellular devices have become important in our daily activities and provided mobile users access to various content anytime and anywhere. Since the content demand of human beings in many cases is location-dependent and time-dependent (spatial-temporal) [20], there are many real world scenarios where co-located cellular users request similar content at approximately the same time.

The first class of scenarios occurs when co-located people attend a jointly co-located event such as a football match in a crowded stadium (as shown in Figure 1(a)). In this case, the co-located fans may use their cell phones to download the same video of a replay scene or the team profile from the content server. This flash-crowd like event incurs a high load on the cell tower; and thus, downloading the video may take from 5 to 10 minutes. The fans, while downloading the video via cellular links, can share downloaded data via the ad hoc interfaces of their cell phones. By doing so, these fans can reduce both their downloading times and the load on the cell tower.

The second class of scenarios can be found when co-located people request information about an publicly popular event as shown in Figure 1(b). For example, when waiting for their buses in the morning at a bus station, mobile users may want to download the breaking news video about an emergency event occurring the night before in Chicago downtown from the *CNN* website. Since it may take from 5 to 15 minutes to download the video from the *CNN* content server via the cellular link, the ad hoc interfaces of these co-located mobile devices can be used to exchange downloaded chunks of data; and thus speed up the download process.

Figure 1(c) shows the third class of scenarios when a group of co-located people is moving toward the same place of a social event such as an outdoor concert. Right before the beginning of the concert, more and more audiences walk toward the concert area, they may download videos of the music trailer and artist profiles from a content server to their cell phones. Again, it may take from 5 to 15 minutes to download the video. Therefore, the ad hoc connections become useful to exchange downloaded chunks of data and accelerate the download process.

The above scenarios commonly occur in our daily activities and naturally create groups of co-located mobile users whose content demands are similar. In reality, these co-located users may not download the similar content at *exactly* the same time. However, since the content in

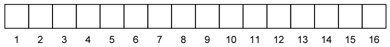


Figure 2. A file consists of multiple equal-sized segments

the above scenarios is spatial-temporal, the downloading periods of co-located users highly *overlap*. Moreover, since human beings are rational in nature, the two co-located users may not always exchange downloaded data although they are simultaneously downloading the same content. As a result, the data exchange among mobile users via ad hoc connections should be considered optional rather than mandatory.

In this paper, we design an Incentive-based Sharing (iShare) protocol to exploit the co-location and mutual content interest of mobile users. Specifically, iShare forms an ad hoc mesh of co-located mobile users and *opportunistically* exploits ad hoc connections among them to exchange data. Since the ad hoc communications among mobile users are opportunistic, the iShare protocol is considered "opportunistic protocol".

2.2 Data Model

In our context, mobile users download a file from the content server. We assume that the file is similar to the Bittorrent [5] file, which has a unique *file id* and consists of multiple equal-sized segments as shown in Figure 2. Each segment also has a unique *segment id* to distinguish it from other segments of the same file. In Figure 2, the file has 16 segments indexed from 1 to 16, each blank square represents a data segment.

With this file format, Bittorrent file servers can be utilized to provide files to mobile users in our network; thus minimizing the changes when applying iShare into the current network infrastructure. If the Bittorrent file servers are not employed, we may need to add a program to the file server to divide files into equal-size segments.

2.3 Network Model

Figure 3 shows our network with a cellular base station and co-located mobile users (or mobile nodes). We assume that a mobile node downloads data from a content server in the Internet via the cellular link and can simultaneously exchange data with other co-located nodes via the ad hoc channel. We assume that mobile nodes in our network have enough power for their ad hoc communications during their downloads. In Figure 3, users (nodes) 1,2,3 download file 1 from server 1 via their cellular links (likewise, users 4,5,6 download file 2 from server 2). We denote user i ($1 \leq i \leq 6$) in Figure 3 "iShare user" since he uses the iShare protocol to download and exchange data. Henceforth, we use the terms iShare user and iShare node interchangeably. In Figure 3, users 8 and 9 don't use the iShare protocol, they use only cellular links to download data, which differs from file 1 and file 2. We denote them "background nodes".

In Figure 3, since iShare nodes 1,2,3 are co-located, they naturally form a downloading group G_1. In one cellular cell, there might exist many downloading groups (e.g., G_1 and G_2 in Figure 3). Nodes in one group form a mesh structure, in which each mesh member may downloads different segments in parallel and they can ex-

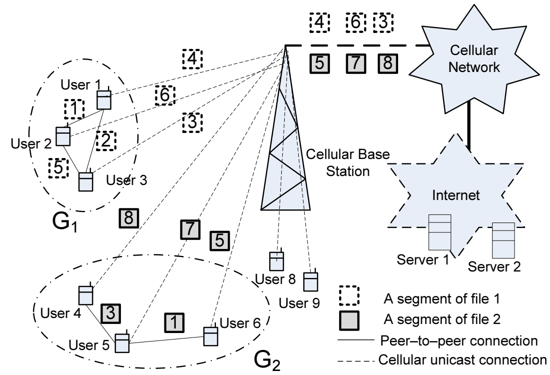


Figure 3. Network Model

change download segments via ad hoc peer-to-peer connections. For example, in Figure 3 node n_1 may download segment 4 while node n_2 may download segment 6, and they are exchanging segment 1. This minimizes the redundant download from the content server and reduces the load on the cellular base station.

2.4 Design Objectives

The iShare protocol has several objectives. First, iShare needs to reduce the downloading time of co-located mobile nodes, under dense networks (Figures 1(a) and 1(b)) or dynamic networks (Figure 1(c)). Second, iShare needs to minimize network overhead to reduce network contention and save node battery. Third, iShare should contribute to improve performance of background users and reduce load on the cell tower. Finally, iShare needs to motivate a fair collaboration and limit the selfishness among mobile users.

3 iShare: Incentive-based Sharing Protocol

3.1 Overview of iShare

The idea of iShare comes from an intuitive observation: human beings are rational and they only collaborate when they find benefits in the collaboration. Therefore, the mutual content interest becomes a key to leverage the cooperation among co-located mobile users. Particularly, iShare nodes continuously download missing segments from the cellular link and at the same time exchange downloaded segments via the ad hoc connections.

In this paper, our iShare protocol mainly focuses on the content layer and control plane of the iShare node architecture as shown in Figure 4. Particularly, an iShare node uses the cellular stack to download segments from cellular link and ad hoc stack to exchange data with ad hoc neighbors. The control plane manages file metadata (e.g., *file id*) and the list of ad hoc neighbors. Figure 5 shows the protocol state machine of an iShare node n, in which n's state depends on n's current activity. First, n stays in states 1,2 while n obtains file metadata and downloads a random segment from content server via the cellular stack in Figure 4. Then, n stays in states 2,3,6 and continuously downloads its missing segments via the cellular stack. At the same time, n attends a mesh ad hoc network of iShare nodes, advertises its available segments, requests its missing segments, and receives seg-

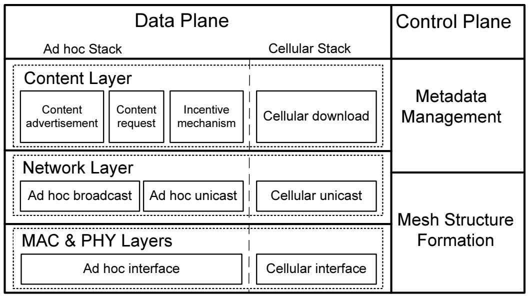


Figure 4. **An iShare node architecture**

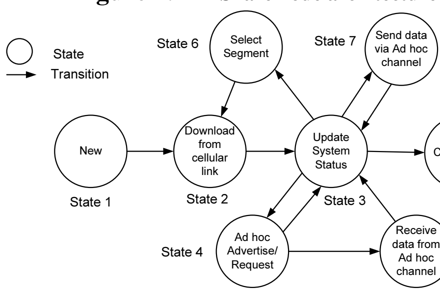


Figure 5. **An iShare node's protocol state machine**

ments from its neighbors by using the ad hoc stack in Figure 4. Here, n stays in states 3,4,5. For a fair collaboration with other iShare nodes, n applies the "tit-for-tat" incentive mechanism to send segments to neighbors via unicast ad hoc links while n stays in states 3, 7. Whenever a segment is obtained by either cellular or ad hoc stack, n switches to state 3 and updates the system status. When obtaining the entire file, n switches to the completed state. Here, due to its rational behavior, n stops all ad hoc communications and collaborations. In the following sections, we present the iShare protocol in detail.

3.2 Bootstrapping the download

This section focuses on states 1,2,3 in Figure 5. When a mobile user starts requesting a file, his cellular device (i.e., the iShare node n) is in the New state. n first obtains the metadata of the file such as *file id* and file length (number of segments) from the content server. The metadata then is managed by the *Metadata Management* module in Figure 4. Initially, this module creates a *segment id list* to hold the ids of downloaded segments. Next, n downloads a random segment s of the file from the cellular link. Receiving s, n stays in state 3 where n puts s into its memory and the *Metadata Management* module inserts the *segment id* into the *segment id list*.

3.3 Ad hoc data exchange

After downloading the first segment from the cellular link, new iShare nodes start combining cellular and ad hoc communications. This section presents ad hoc communications among co-located nodes and focuses on states 3,4,5 in Figure 5.

3.3.1 Content advertisement and request

For an iShare node n, the ad hoc channel is used to advertise its available segments and request its missing segments. Particularly, n periodically broadcasts a HELLO

0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 6. **The HELLO message format**

message, which is in bit vector format as shown in Figure 6. In this figure, the index of the bit, starting from 1 to 16, is the *segment id*; for example, the 12^{th} index denotes the 12^{th} segment. The HELLO message represents the latest downloaded segments of n and is created from the *segment id list*. Notice that the length of a HELLO message is the file length or number of segments. The HELLO message can be used as both segment advertisement and segment request, where 1 represents one downloaded segment and 0 represents a missing segment in n's memory. Thus, the HELLO message efficiently reduces ad hoc network contention.

To further reduce the overhead and save battery, especially when there are no ad hoc neighbors, node n can turn on the ad hoc interface and broadcast HELLO messages to find neighbors for a certain period (e.g., 10 seconds). After this period, if n does not receive any HELLO messages from its neighbors, n turns off ad hoc interface for another period, and repeats the whole process periodically. If n receives one HELLO message from neighbors, n keeps the ad hoc interface on to exchange data with these neighbors. Since it takes from 5 to 10 minutes to download a file via cellular links, two co-located cellular users are likely to find each other via HELLO messages.

3.3.2 Mesh structure formation and data exchange

Upon receiving a HELLO message, a node n keeps its ad hoc interface on to exchange data with the neighbors. The *Mesh Structure Formation* module in Figure 4 keeps the list of current ad hoc neighbors, which is used to from a mesh structure of nodes in a downloading group as shown in Figure 7. The mesh has following characteristics. First, the mesh structure is formed automatically since the co-located nodes are within the ad hoc communication range and thus incurs little construction/maintenance cost since nodes only need to keep the one-hop neighbor list [1]. Second, any two mesh neighbors can exchange data if they are within communication range. However, a node may need to download data from cellular link at anytime due to node mobility and user rational behavior (user may stop ad hoc communications when finishing his download). Finally, each mesh member communicates directly with its one-hop neighbors. This one-hop communication adapts well with network dynamics and fits very well with our tit-for-tat incentive mechanism in Section 3.5.

Two mesh neighbors n_1 and n_2 can exchange HELLO messages for segment advertisements and requests. Upon receiving HELLO messages from n_1, n_2 knows the missing segments of n_1. Then, n_2 applies the tit-for-tat incentive mechanism to send these segments to n_1. At the moment, n_1 stays in state 5 in Figure 5 and receives the segments. If the received segment s is new to n_1, then n_1 inserts s into its memory and n_1's *Metadata Management* module updates the *segment id list* accordingly.

[1] HELLO message is the overhead of data advertisement

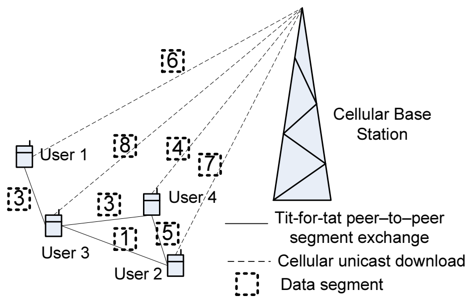


Figure 7. **A mesh network of iShare nodes**

3.4 Downloading data from cellular link

As presented above, iShare nodes exchange downloaded data via ad hoc connections. At the same time, they continuously download segments from the cellular links. In this section, we present how iShare nodes download segments from cellular links and we focus on states 2,3,6 in Figure 5.

3.4.1 Segment Selection

Since the ad hoc connection is opportunistic, iShare nodes uses their cellular stacks in Figure 4 to download segments from the content server via cellular links. To reduce the download from the content server and the load on the base station, iShare nodes utilize HELLO messages to download the best segment.

Specifically, node n downloads its missing segment s, the least available segment in its neighborhood. Thus, when downloaded, s can support file downloading among ad hoc nodes most effectively. Particularly, n aggregates all the latest HELLO messages received from its one-hop neighbors to create an aggregated HELLO message as shown in Figure 8. In this figure, each square can be a byte instead of a bit like the HELLO message. Notice that for each neighbor n_1 of n, n only keeps the latest HELLO message received from n_1 for the most updated available segments of n_1. In Figure 8, a square represents a segment s with the number of available copies of s in n's neighborhood. Node n downloads its missing segment s whose number of copies is least founded in the aggregated HELLO message. If there exist more than one missing segments with equal number of copies, n downloads one at random. For example, if n's sent HELLO message is in Figure 6 and n's aggregated HELLO message is in Figure 8, then n may download segment 3 from the cellular link since segment 3 is missing at n and n's neighbors. Whenever n finishes downloading a segment s, n is in the state 3 in Figure 5. Here, n inserts s into its memory and the *Metadata Management* module inserts s's id into the *segment id list*. Then, n continues downloading its missing segments from the cellular link.

3.4.2 Parallel download from the cellular link

The random segment selection presented above parallelizes the download among mesh members. Particularly, mesh members concurrently download different segments and exchange these segments via ad hoc connections as shown in Figure 7. The aggregated HELLO message thus minimizes redundant downloads from the content server and reduces the load on the base station.

2	0	0	4	1	3	0	0	0	3	0	4	3	0	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 8. **The aggregated HELLO message format**

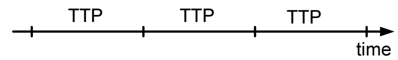


Figure 9. **Time is divided into Tit-for-Tat Periods (TTP)**

3.5 The tit-for-tat incentive mechanism

The mesh ad hoc structure efficiently parallelizes the download and reduces the load on the cell tower. However, for an efficient iShare protocol, we need to design an incentive mechanism to motivate the collaboration of iShare nodes. Therefore, we present the tit-for-tat incentive mechanism to motivate node collaboration. Here, we focus on states 3 and 7 in Figure 5.

3.5.1 Applying the tit-for-tat

Human beings are rational in nature and in most cases they only collaborate if they find benefit in the collaboration. Thus, we exploit the mutual content interest of co-located mobile users to design a simple, yet practical, incentive mechanism. Particularly, our incentive mechanism comes from the "tit-for-tat" method [5] in Game Theory. For two iShare nodes n_1 and n_2, tit-for-tat means if n_1 gives c segments (needed by n_2) to n_2 then n_2 will give c segments (needed by n_1) to n_1.

Applying tit-for-tat, iShare nodes divide time into equal-sized periods, called tit-for-tat periods (TTPs) as shown in Figure 9. The TTP is then used as follows. Given two ad hoc neighbors n_1 and n_2. Node n_1 uses the current TTP to receive segments from n_2 so that n_1 can send n_1's segments back to n_2 in the next TTP. Also, n_1 sends segments to n_2 in the current TTP so that n_2 can send segments to n_1 in the next TTP. Notice that the length of a TTP is longer than that of the HELLO message broadcast period since nodes need to update available/missing segments to effectively perform tit-for-tat.

For a fair collaboration, during a TTP, an iShare node n counts the number of segments received from its neighbors. Given two neighbors n_1 and n_2, during a TTP, if n_2 sends 15 segments to n_1, then n_1 has counter $c_2 = 15$, corresponding to n_2. Notice that the counter is updated at the state 3 in Figure 5. At the end of the TTP, n_1 is at the state 7, if n_1 has more than 15 segments that n_2 needs (known from n_2's HELLO message), n_1 only sends 15 segments to n_2 via *unicast ad hoc link*. If n_1 has less than 15 segments needed by n_2, n_1 sends all to n_2. The unicast ad hoc link is to motivate the fair collaboration between neighbors and ensure reliable ad hoc data exchange. In the next section, we present how to bootstrap and adapt the tit-for-tat for the dynamic network.

3.5.2 Bootstrapping and Improving the tit-for-tat

The tit-for-tat mechanism presented above encourages iShare nodes to collaborate. However, it may not work effectively if the node neighborhood changes swiftly, since a new pair of neighbors need to bootstrap tit-for-tat from scratch. Thus, we present two techniques to bootstrap and adapt the "tit-for-tat" to network dynamics.

First, we turn on the promiscuous mode so that the iShare node n can potentially overhear messages, which are destined to n's neighbors in ad hoc unicast communication of the tit-for-tat. By doing so, n can opportunistically receive more data from ad hoc channel. Of course, when the network is dense or congested, the overheard messages can be dropped or collide and n misses the chance. Second, during a TTP, n broadcasts in the ad hoc channel a small number of its segments whose available copies are least in n's neighborhood. The segment availability is obtained from the aggregated HELLO message.

Using the promiscuous mode and broadcast mechanism, iShare nodes improve the "tit" step of the tit-for-tat so that they exchange more segments in the "tat" step. These two techniques allow the new neighbors to exchange data effectively without restarting the tit-for-tat from scratch. These techniques also enable new nodes to join the downloading group smoothly since they are given several segments for free. However, to exchange data with the old nodes effectively, new nodes need to download new segments from the content server. Otherwise, they become "iShare selfish nodes" and their downloading times may last longer as explained below.

3.5.3 iShare selfish nodes

In our context, an iShare selfish node n wants to have fast downloading time and minimize power consumption by only overhearing ad hoc messages from the ad hoc channel. To this end, n downloads segments from cellular link and turns on the ad hoc channel just for overhearing free segments from promiscuous mode and ad hoc broadcast channel. In other words, n does not actively send HELLO messages and exchange data with other nodes via ad hoc connections. Relying on these free segments; however, n has no guarantee to receive its missing segments from the ad hoc channel, since the network contention and packet collision may drop the overheard messages. Thus, n fails to utilize the ad hoc channel and thus suffers a longer downloading time (see Figure 12(b)).

Whenever an iShare node n receives a new segment from either cellular or ad hoc link, n stays in state 3 in Figure 5 and updates its system status. Then, n checks whether it finishes downloading the entire file. If so, n switches to the completed state.

3.6 Completed state

At this state, an iShare node n leaves the mesh ad hoc network and stops all ad hoc communications since n has no incentive to support other downloading nodes. This is because n is rational and n only collaborates when n is downloading the file.

4 Evaluation

We present simulation settings and simulation results of iShare for three cases: a single downloading group in one cell, multiple downloading groups in one cell, a downloading group spanning over two adjacent cells.

4.1 Simulation Settings

We use Network Simulator 2 (NS2) to simulate a cellular cell and mobile nodes with the settings in Table 1. In this table, the segment size is 4KB since smaller segment incurs longer HELLO message while bigger segment may cause more ad hoc collision. If the content server is a Bittorrent server, we can always add a function to divide Bittorrent segments into 4KB segments. In our simulation, a node has two interfaces: one is for the cellular link and the other is IEEE 802.11b for the ad hoc communication. We use RTS/CTS for unicast ad hoc communications. For the cellular technology, we use 1xEV-DO (Evolution-Data Only) with a peak data rate of 2.4Mbps. We also implement the Proportional Fair Scheduler of the cellular network [2] in NS2.

Field	**Value/Unit**
Number of background nodes	[10,20,30,40,50]
Downloading group size	[10,15,20,30,40]
Segment size	4KB
File size	[1000...6000] segments
Node ad hoc transmission range	125(m)
Base station radius	750(m)
Mobility model	Random Way Point
Node speed (Mobility-NS2)	[1,3,5,7,11] (m/s)
Pause time (Mobility-NS2)	5 (seconds)

Table 1. Simulation settings

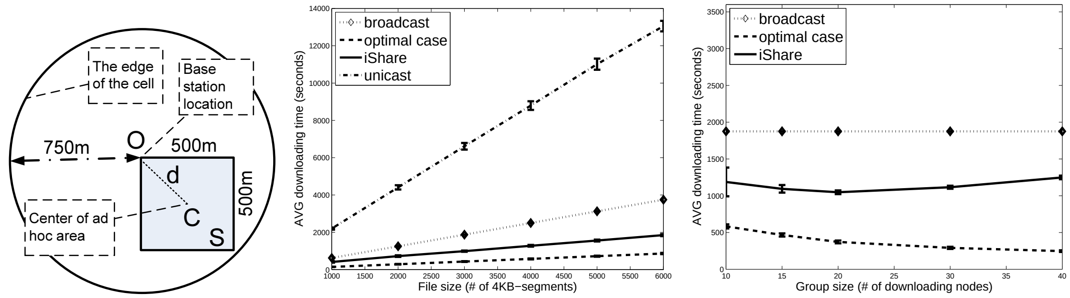
Our simulation considers iShare nodes and background nodes (see Section 2.3 for definitions). We evaluate two metrics: "average iShare node downloading time" and "average number of downloaded segments by background nodes". The former is the average (AVG) period for an iShare node to finish downloading a file. The latter is the average number of segments a background node can download via *only* the cellular link for a given period.

In our plots, "optimal case" means the time at which the first copy of all segments of the file are downloaded by iShares nodes from the cellular link. At the moment, each iShare node may hold a portion of the file but they need to exchange via ad hoc connections so that each node can retrieve the entire file. We assume an oracle exists to magically deliver these segments to all iShare nodes immediately. Thus, the optimal case approximates the lower bound of the downloading time of the iShare nodes. We also compare iShare with cellular "broadcast" and "unicast" channels. Cellular broadcast channel means the base station broadcasts the file to all iShare nodes with a fixed rate of 208.4 Kbps. That means, iShare nodes don't use ad hoc channel, they only use cellular broadcast channel. Here, we assume that 25% of cell bandwidth is for the broadcast channel and 75% is for background nodes. Cellular unicast channel means iShare nodes only use cellular unicast link to download and iShare nodes compete with background nodes for 100% of cell bandwidth.

For iShare, a HELLO message is broadcast every 2 seconds. The tit-for-tat period TTP is 7 seconds and the number of broadcast (tit) segments is 1% of the file size. The default configurations of our plots are: group size is 15, node speed is 5 (m/s), number of background nodes is 30, file size is 3000 4KB-segments. We specify when the parameters are different. We run each simulation 10 times and plot the mean with the 95% confidence interval. Since the unicast channel always performs worst, its result is plotted when appropriate.

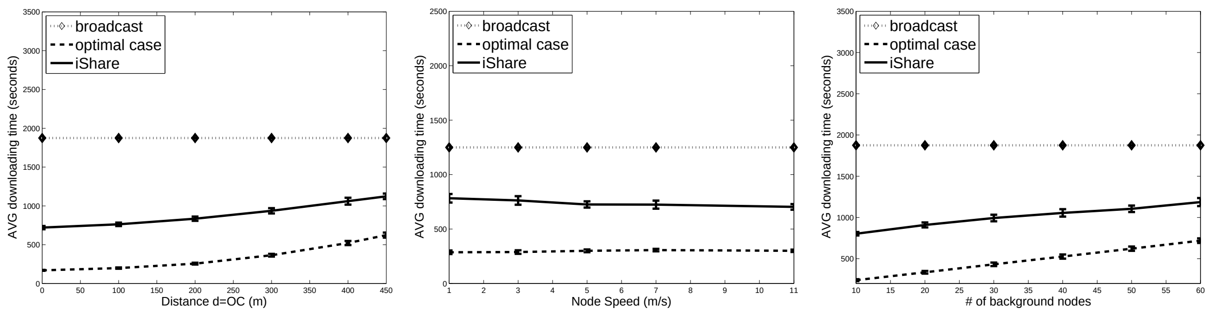
4.2 A Single Downloading Group

In this section, we focus on a single downloading group in one circle-shaped cellular cell as shown in Fig-



(a) Experiment Area (b) Impact of File size (c) Impact of Group size

Figure 10. Performance of a single downloading group in a cell



(a) Impact of distance to base station (b) Impact of network dynamics (c) Impact of background traffic

Figure 11. Performance of a single downloading group in a cell (2)

ure 10(a). In this figure, the base station is at *O* and all nodes are generated at random inside the square *S*.

4.2.1 Impact of file size

Figure 10(b) shows that the average downloading time of iShare nodes linearly increases when the file size increases. The optimal case has lowest downloading time and iShare outperforms both broadcast and unicast channels. Moreover, for the larger file size, iShare constantly achieves shorter downloading time.

4.2.2 Impact of group size

Figure 10(c) shows that for the bigger downloading group, although the downloading time of iShare nodes slightly increases, iShare is always better than broadcast channel. When the group size is 10, the ad hoc network is too sparse to be exploited efficiently, resulting in higher downloading time of iShare. When the group size increases more than 30, the downloading time of iShare increases slowly since ad hoc communications improve performance of iShare nodes. This slightly decreased performance also confirms that the use of HELLO messages reduces ad hoc contention significantly.

4.2.3 Impact of distance to the base station

For the cellular network, in most cases the distance from the cell phone to the base station influences the channel condition and downloading rate of the cell phone. Therefore, the distance becomes important to iShare nodes. Let d be the distance from the base station to C, the center of S, $d = OC$ in Figure 10(a). We vary d by moving S along with OC, starting from O toward C. Figure 11(a) shows that when d is shorter (more nodes close to the base station), iShare obtains shorter downloading time (a factor of 3) than that of broadcast channel. When d becomes longer, more low-data-rate nodes exist, the downloading time slightly increases. When $d = 450$, we have a significant number of low-data-rate nodes at the edge of the cell, iShare still significantly outperforms broadcast channel since the mesh structure parallelizes download among iShare nodes to speed up the download. This result confirms that iShare works very well with low-data-rate nodes.

4.2.4 Impact of network dynamics

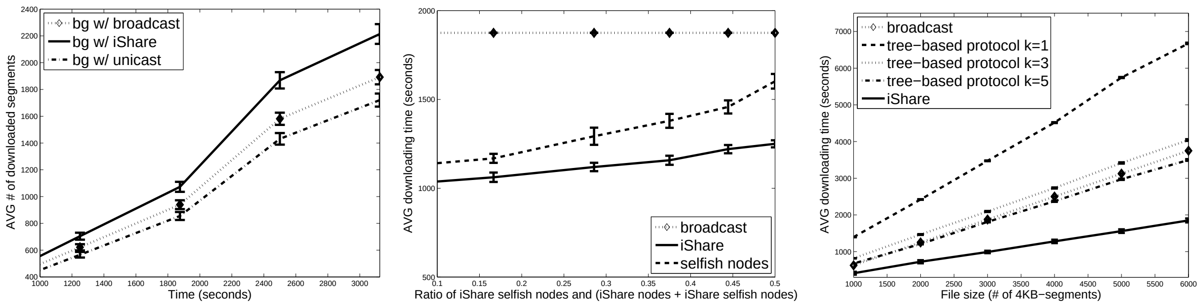
In this section we study the performance of iShare under the change of node neighborhood. When nodes move faster, node neighborhoods change faster and this influence iShare nodes. For this simulation, the file size is 2000 segments. Figure 11(b) shows when node speed increases, the performance of iShare remains stable and always better than cellular broadcast channel. This confirms the robustness of iShare to network dynamics.

4.2.5 Impact of background traffic

Figure 11(c) shows that when the number of background users increases, the performance of iShare slightly and linearly degrades due to the increased load on the cellular link. However, iShare is always better than broadcast channel. This figure also denotes that the increase of background nodes incurs little effect on iShare since the ad hoc channel can accelerate download of iShare nodes.

4.2.6 Performance of background nodes

Figure 12(a) shows that when iShare nodes use iShare protocol to download data, the background nodes (i.e. bg w/ iShare) can obtain more data than when iShare nodes



(a) Performance of background nodes (b) Performance of iShare selfish nodes (c) A tree-based protocol vs. iShare

Figure 12. Performance of a single downloading group in a cell (3)

are served by cellular broadcast and unicast channels (i.e., bg w/ broadcast and bg w/ unicast) (see Section 4.1 to see how cellular broadcast/unicast channels are used to support iShare nodes.) For a longer time period, "bg w/ iShare" consistently shows better performance. This is because when mobile nodes use iShare protocol, they finish their downloads faster than when the mobile nodes are served by cellular broadcast/unicast channels; thus, the background nodes can use more cell bandwidth to download more data. iShare, therefore, creates a win-win situation for both downloading and background nodes. This result is further confirmed in Section 4.3.2.

4.2.7 Performance of iShare selfish nodes

The definition of iShare selfish nodes can be found in Section 3.5.3. In this simulation, we have 15 iShare nodes and we vary the number of iShare selfish nodes from 1 to 15. Figure 12(b) shows that when more iShare selfish nodes exist, their downloading times increase noticeably. Meanwhile, the downloading times of iShare nodes only increase slightly due to the higher load on the cell tower. In other words, iShare selfish nodes suffer from their own existences and iShare thus limits the selfishness.

4.2.8 A tree-based protocol vs. iShare

This section compares the performance of iShare and a tree-based protocol of cooperative downloading nodes, which is implemented as follows. Nodes periodically update their distances to the base station and elect the proxies whose distances to the base station are shortest (implying the best channel conditions). These proxies download segments from the base station and broadcasts the segments to tree members, which replay the segments through the tree of ad hoc nodes. When a proxy *p* finishes its download, *p* notifies the group so that a new proxy is elected. At the same time, *p* keeps its ad hoc connection on to support other nodes. We make sure the group is always connected. Here, iShare and the tree-based protocol both use the same simulation settings.

Figure 12(c) shows that iShare protocol consistently outperforms the tree-based protocol. In this figure, k denotes number of concurrent proxies. When $k = 1$, the tree-based protocol performs much worse than cellular broadcast channel. When k increases, the tree of multiple roots performs noticeably better, although always worse than iShare. Figure 12(c), together with Table 2, confirms that iShare owns more advantages than tree-based protocols in improving performance of mobile nodes. In Table 2, the overlay overhead of iShare is "Negligible" since the overhead of HELLO message is counted for data advertisement (Section 3.3.2).

Property	Tree-based protocol	iShare
Data overhead	Data advertisement, data exchange	Data advertisement, data exchange
Overlay overhead	Root selection, tree construction/maintenance	Negligible
Full cooperation required	Yes	No (Tit-for-tat)
Network dynamics	Vulnerable	Not vulnerable

Table 2. **Comparison between a tree-based protocol and iShare**

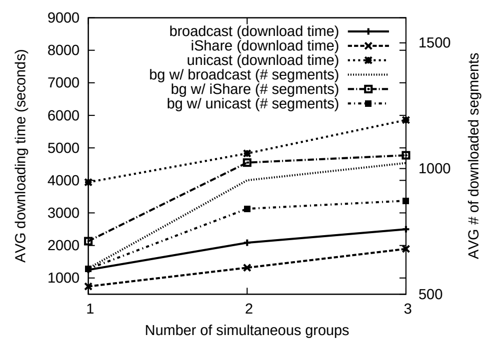


Figure 13. **Impact of simultaneous groups in one cell**

4.3 Simultaneous Downloading Groups

4.3.1 Simulation Settings

Here, we use the settings in Table 1 and Figure 10(a). We have 3 groups within *S*, each group has 10 nodes and downloads a different file; thus, there is no inter-group communication in ad hoc mode. We assume that cell bandwidth for the broadcast channel is 25%, 15%, 12.5% corresponding to 1, 2, 3 simultaneous groups; or 25%, 30%, 37.5% aggregated bandwidth for the broadcast channel. Here, the file size is 3000 segments, node speed is 5 (m/s), number of background users is 30.

4.3.2 Impact of simultaneous groups

Figure 13 shows that the downloading time of iShare nodes increases when more groups exists, since iShare nodes suffer from a higher contention in the ad hoc channel. Moreover, due to the promiscuous mode, iShare nodes receive redundant messages from nodes in other groups, which may collide with the desired overheard messages from nodes in the same group. As a result,

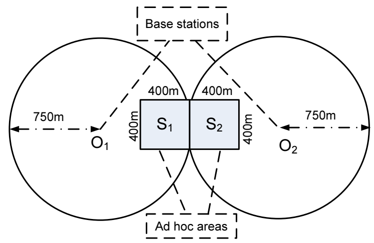


Figure 14. **Experiment area of one spanning group**

the performance of tit-for-tat degrades. However, iShare performs consistently better than broadcast channel. Figure 13 also shows background nodes download more data (right y-axis) from cellular links if mobile nodes use iShare protocol (i.e., bg w/ iShare). Similar to the Section 4.2.6, if downloading nodes use iShare, they finish the downloads faster; thus background nodes can use more cell bandwidth to download more data. So, we confirm that iShare creates a win-win situation by improving the performance of both downloading and background nodes.

4.4 A Spanning Group

4.4.1 Simulation Settings

In a cellular cell, mobile nodes, which are close or at the edge of a cell, usually have bad channel conditions with very low data rates. This section evaluates the performance of iShare for these low-data-rate nodes. We first create a group g_1 of 10 iShare nodes within the square S_1 in Figure 14. Then, we create a group g_2 of 20 iShare nodes within two squares S_1 and S_2 spanning two adjacent cells centered at O_1 and O_2. We call g_2 a spanning group, which has a significant number of nodes close and at the edge of two adjacent cells. Nodes in g_1 and g_2 download segments from their current base stations and exchange segments via the ad hoc channel. Here, node speed is 7 m/s and each cell has 30 background users.

4.4.2 Performance of a spanning group

Figure 15 shows that iShare with g_2 (2 cells case) consistently outperforms iShare with g_1 and broadcast channel, especially for the larger file size. This result has several implications. First, iShare provides an efficient method to reduce downloading time, especially for low-data-rate nodes at the boundary of the cellular cell. Second, iShare nodes can continuously obtain data via the ad hoc channel during their cellular handoff periods. Finally, iShare offers the multi-homing download for a spanning group, where group members download content from different/adjacent base stations and exchange segments via ad hoc connections to improve downloading throughput.

5 Related Work

There have been previous works on the combination of cellular and ad hoc networks to improve downloading bandwidth of mobile users [1][10][12][11][14][16][18]. These approaches put new functionalities on the cellular telephony infrastructure such as a new scheduler, membership management, credit verification; and thus require a high cost of deployment. In particular, these approaches select the high-data-rate nodes such as proxies or super nodes to connect to the base station. They next construct and maintain trees rooted at proxies. Then, the packets are sent from the base station to the proxies and forwarded to the receivers, assuming that nodes are collaborative. Under network dynamics, maintaining these trees incurs high overhead. If the proxies leave the cell, the trees need to be reorganized. In contrast, iShare requires no changes in the cellular telephony infrastructure since network functionalities are performed by ad hoc nodes.

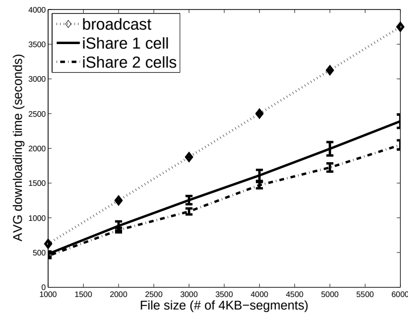


Figure 15. **One downloading group spanning two adjacent cells**

Another possibility to provide content to simultaneous users is using the cellular broadcast/multicast channel [6][8][9][13][17]. However, multicast/broadcast services [8][9] are to support content to a large number of users simultaneously; thus, the number of multicast/broadcast channels is usually limited [9]. Meanwhile, in reality, people can instantly form small-scaled groups to exchange content; broadcast/multicast services thus become inefficient. Moreover, multicast/broadcast services have no feedback channel, the data is delivered with no guarantee. There also exists work combining cellular unicast with ad hoc links to improve the downloading bandwidth of multicast users by creating a tree rooted at a proxy to relay packets [3]. However, this approach suffers a high cost of tree maintenance under network dynamics.

Previous studies on wireless networks presented numerous incentives mechanisms to motivate the collaboration of mobile users such as market sharing, credit accounting, and rewarding [4][7][15][19][21]. However, these mechanisms add significant complexity to the system and applying them requires changes in the business policy of service providers. Moreover, due to the scarcity of battery, there is no immediate incentive for a user to turn on his ad hoc channel just for forwarding others' data. Our tit-for-tat mechanism is simple yet practical because they reflect the rationale of human beings: co-located mobile users are willing to collaborate if they share the mutual content interest.

6 Conclusions

In this paper, we have developed an Incentive-based Sharing (iShare) protocol that combines cellular and ad hoc connectivity on mobile devices to improve content dissemination services to mobile users. Thereby, users download content from a source via cellular links and at the same time leverage ad hoc connections to other users for exchanging content chunks following the peer-to-peer paradigm. The mesh formed with other nodes remains very robust to network dynamics and minimizes ad hoc communication overhead. Furthermore, it parallelizes the downloading process among mesh members, which yields accelerated download performance for the users and at the same time reduced load on the cellular network. Our solution includes an efficient and practical the "tit-for-tat" incentive mechanism that helps the cooperation amongst members. By utilizing promiscuous and broadcast modes of the ad hoc channel, our incentive mechanism is particularly suited and effective in mobile scenarios with network dynamics.

We have implemented our protocol in simulator and performed an extensive performance study. The obtained results confirmed our hypothesis and showed that iShare significantly outperforms alternative schemes based on cellular broadcast channels, cellular unicast channels, or tree-based protocols. Furthermore, in the evaluation study our protocol demonstrated how the implemented incentive mechanisms succeeds in countering selfishness user behavior. The experiments also showed that users continue obtaining data via ad hoc connections during the cellular handoffs, when the cellular connection is not available or very poor. Finally, the obtained results confirmed the multi-homing download feature for groups spanning over adjacent cellular cells.

There are numerous scenarios in reality where co-located cellular users demand similar content at the same time window. For these scenarios, the iShare protocol provides very efficient and practical mechanisms for acceleration of content dissemination to mobile users and reduction of cellular network load. Toward this end, iShare is novel and widely applicable.

7 References

[1] S. G. Aly and A. A. Elnahas. Sustained service lookup in areas of sudden dense population. *Wireless Communications and Mobile Computing*, 8:61–74, 2008.

[2] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi. CDMA/HDR: A bandwidth-efficient high-speed wireless data service for normadic users. *IEEE Communications Magazine*, 2000.

[3] R. Bhatia, L. E. Li, H. Luo, and R. Ramjee. ICAM: Integrated cellular and ad-hoc multicast. *IEEE Transactions on Mobile Computing*, 5(8):1004–1015, August 2006.

[4] K. Chen, Z. Yang, C. Wagener, and K. Nahrstedt. Market models and pricing mechanisms in a multihop wireless hotspot network. In *Mobiquitous*, 2005.

[5] B. Cohen. Incentives build robustness in BitTorrent. In *IPTPS*, 2003.

[6] A. Correia, J. Silva, N. Souto, L. Silva, A. Boal, and A. Soares. Multi-resolution broadcast/multicast systems for MBMS. *IEEE Transactions on Broadcasting*, 53:224–234, 2007.

[7] M. Goemans, L. Li, V. Mirrokni, and M. Thottan. Market sharing games applied to content distribution in ad hoc networks. *IEEE Transactions on Mobile Computing*, 24:1020–1033, May 2006.

[8] D. Gomez-Barquero, A. Aguilella, and N. Cardona. Multicast delivery of file download services in 3g mobile networks with MBMS. In *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2008.

[9] F. Hartung, U. Horn, J. Huscheke, M. Kampmann, T. Lohmar, and M. Lundevall. Delivery of broadcast services in 3G networks. *IEEE Transactions on Broadcasting*, 53, 2007.

[10] H.-Y. Hsieh and R. Sivakumar. On using peer-to-peer communication in cellular wireless data networks. *IEEE Transactions on Mobile Computing*, 3, 2004.

[11] I. Ioannidis, B. Carbunar, and C. Nita-Rotaru. High throughput routing in hybrid cellular and ad-hoc networks. In *World of Wireless Mobile and Multimedia Networks WoWMoM*, 2005.

[12] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla. Bluetorrent: Cooperative content sharing for bluetooth users. *Pervasive and Mobile Computing*, 3:609–634, 2007.

[13] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson. Reliable multimedia download delivery in cellular broadcast networks. *IEEE Transactions on Broadcasting*, 53:235–246, 2007.

[14] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. The design and evaluation of unified cellular and ad hoc networks. *IEEE Transactions on Mobile Computing*, 6:1060–1074, 2007.

[15] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *NSDI*, 2005.

[16] J. C. Park and S. Kasera. Enhancing cellular multicast performance using ad hoc networks. In *Wireless Communications and Networking Conference WCNC*, 2005.

[17] S. Parkvall, E. Englund, M. Lundevall, and J. Torsner. Evolving 3G mobile systems: Broadband and broadcast services in WCDMA. *IEEE Communications Maganize*, 2006.

[18] K. S. and M. M. Efficient mobile access to internet data via a wireless peer-to-peer network. In *Pervasive Computing*, 2004.

[19] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson. Charging and rewarding scheme for packet forwarding in multihop cellular networks. In *Mobihoc*, 2003.

[20] L. Vu, K. Nahrstedt, and M. Hollick. Exploiting schelling behavior for improving data accessibility

in mobile peer-to-peer networks. In *Mobiquitous*, 2008.

[21] W. Wang, X. Y. Li, and Y. Wang. Truthful multicast in selfish wireless networks. In *Mobicom*, 2004.