Sangam - Efficient Cellular-WiFi CDN-P2P Group Framework for File Sharing Service

Anjali Sridhar
University of Illinois,
Urbana-Champaign
Urbana, USA
sridhar3@illinois.edu

Klara Nahrstedt
University of Illinois,
Urbana-Champaign
Urbana, USA
klara@illinois.edu

Long Vu
ByteMobile
Urbana, USA
longvu2@illinois.edu

ABSTRACT

Solutions to the cellular network bandwidth problem have been presented by the community such as usage of alternative wireless (e.g., 802.11 WiFi) network and employ peer-to-peer (P2P) file sharing over the group of wireless devices. The multi-radio approach for file sharing has been validated analytically or via simulation in the literature. However, often the theoretical and simulation approaches for file sharing within multi-radio CDN-P2P groups hide the complexity of systems and networks in real scenarios such as heterogeneity of phones in a P2P group, issues with scheduling policies within a CDN-P2P group of devices, CDN-P2P group formation and others.

In this paper, we present Sangam, an efficient cellular-WiFi CDN-P2P group framework for file sharing where we address extensively system and network challenges in file sharing for real phone group scenarios. The Sangam framework accounts in its protocol, policy and algorithmic designs for (a) heterogeneity of phone group devices in terms of CPU and power levels, (b) different sizes and numbers of chunks in P2P part of the group-based content distribution, (c) hybrid scheduling policies for chunk dissemination within multi-radio CDN-P2P wireless group environment, and (d) different group sizes. Sangam validation shows the impact and difference to simulations when considering real implementation of video file sharing within a cellular-WiFi CDN-P2P group.

1. INTRODUCTION

The mobile smartphone industry has revolutionized the way that users use the internet and connect with other users. One of the most popular services provided by smartphones is downloading and playing videos.The majority of the traffic on the Mobile Handheld Devices(MHD) is seen to be multimedia content followed by web browsing.Cisco reports that it expects video traffic to account for 60% of the global IP traffic by 2014 [2]. The mobile devices use the cellular interface to connect to the internet. The network speed offered by current cellular interface is only a few megabytes per second [6]. The time required to download a short video file is in the order of a few minutes.This severely degrades the quality of service provided to the user. Comparison studies in [13] and [7] show efficiency of the Wi-Fi interface in comparison to the cellular interface. The Wi-Fi interface not only provides a higher throughput but also consumes much less power than the cellular interface while downloading large files.

Hence increased growth of mobile phones and their capabilities to access multi-radio wireless infrastructure can lead to extended video file download services using these devices. In places where there is no wired network infrastructure like open concerts, in stadiums or large factories, the nodes use the cellular link to download the file of interest.Examples of such scenarios are viewing the same video file among a co-located group of fans at a concert, or downloading the same class material to co-located group of students in the classroom onto their mobile devices.To enable such group-based scenarios, cellular content distribution networks (CDN) have been used for file download and implicit sharing of content among groups of wireless devices [14] [6]. However, the cellular network bandwidth is becoming saturated, hence file download services are suffering from long delays, losses and even drop in connectivity in co-located group scenarios.

Another very important issue in mobile devices today is the limited battery power available. In [22] we see that the transmission power over cellular networks and Wi-Fi access points require the largest percentage of power consumption in a device. The efficiency as seen in [7] shows us that the Wi-Fi interface consumes considerably less power than the cellular interface especially when transferring large files.

In this paper, we utilize the mobile phone capabilities to access multi-radio(cellular-Wi-Fi) wireless infrastructure and explore solutions to assist co-located group of users to download video files in timely and energy efficient manner, hence yield same quality of service in mobile phones as in laptops. We present the Sangam framework that uses multiple radio interfaces of a mobile group based ecosystem to decrease the time of file download. All mobile devices belonging to a co-located

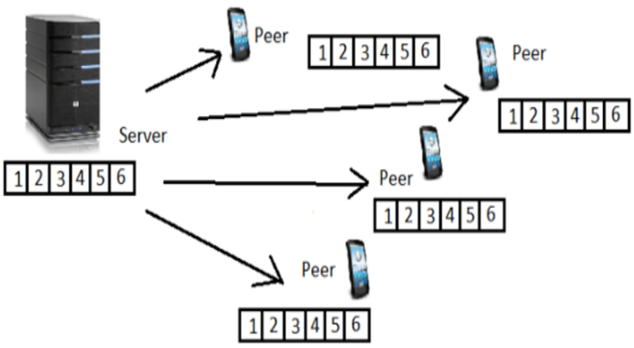


Figure 1: Problem Statement

group are equipped with cellular, Wi-Fi and Bluetooth interfaces and connect directly with other mobile devices in the vicinity.This connectivity between mobile devices leads to the formation of a wireless peer-to-peer(P2P) network. This is what we refer to as a group of peers.

If the group of peers are interested in the same file, duplicate requests are made to the server. The repeated requests for the same file by the co-located group of peers results in unnecessary usage of cellular bandwidth. The Sangam framework leverages the proximity of the co-located group of peers and their interest in a common file to minimize the file download time and cellular bandwidth use. Sangam combines the cellular and Wi-Fi wireless interfaces to create a hybrid CDN-P2P overlay network for group-based file sharing.

Using the Sangam framework we answer the following questions:

1. Does the joint cellular/Wi-Fi hybrid network provide us with a lower download time than with a purely cellular network?
2. What scheduling policies for content dissemination need to be applied? What are the considerations entailing such policies?
3. What is the advantage of accounting for the heterogeneity of devices?
4. How is the Sangam framework helpful in the power management of devices?

Hence, our paper has the following contributions:

1. To the best of our knowledge, this is the first system implementation and experimentation of a hybrid network using multiple radio interfaces.
2. The Sangam framework introduces, evaluates and compares three CDN-P2P coordinating download and distribution policies while taking into consideration the heterogeneity of the peer group of devices.
3. Reduced power consumption is attained using the hybrid network while downloading large files.

The paper is organized as follows. In Section 2 we present some of the related work in this field. Section 3 states our assumption when solving the problem. We present our problem description in Section 4. In Section 5 we describe the proposed Sangam architecture as a viable solution. In Section 6 we present our experimental environment followed by our evaluation of the Sangam framework in Section 7. We present our conclusion in Section 8.

2. RELATED WORK

There have been many hybrid models that have been suggested in order to overcome the challenges faced by a pure client/server model over a single network interface. The hybrid model takes advantage of the fact that there are multiple peers interested in the same file content and uses a peer to peer network to augment the client/server model. Rimac et al [23] develop a mathematical model to evaluate a hybrid model on the basis of churn rate, object size and upload bandwidth. The evaluations show that the hybrid model is sensitive to churn rate but performs better than the traditional model in terms of resource allocation and time convergence.

Ishare [26] envisioned a scenario where there were co-located mobile devices requesting the same file from a server. The file of interest is divided into segments and downloaded by each of the peers over the cellular network. The mobile device also requests file segments from other mobile devices in the peer to peer network. In Ishare design, the download from the server over the cellular interface and the download from the peers over the Wi-Fi interface takes place simultaneously. Each file segment is requested at random and each peer tries to download the maximum number chunks possible in the least amount of time.

There have been many suggestions for using the base station to assist in the scheduling of P2P wireless networks. One of most common approaches, as mentioned by Hsieh et al [16] is to use the base station for deciding the scheduling of the P2P networks which will offer better throughput in terms of less packet loss. The ICAM (Integrated Cellular and Adhoc Multicast) architecture proposed by Bhatia et al [8] puts forth the concept of mobile proxies with the highest downlink rate that are used to forward the data to other mobile peers. The problem of fairness is addressed by UCAN (Unified Cellular and Adhoc Networks. UCAN, proposed by Luo et al [19], introduces the concept of secure crediting. While selfish hosts are not assumed, there is a crediting system that is used to allow proxies and relays transmitting information to be credited for their relaying work. Augmenting the existing cellular network with peer to peer communication is also suggested in Madsen et al

[20]. The paper proposes cooperative network of cellular and P2P networks called Cellular controlled P2P communication networks. The mobile hosts use both the cellular network for communication with the base station and a Bluetooth connection for communicating with peers. An implementation of a file download process on Nokia phones N70 running the Symbian operating system posted decreased energy consumption of about 44% with the download time being half the original time.

Sangam uses multiple interfaces on group-based mobile devices in conjunction with peer-assisted data dissemination. There have been several approaches that have used file segmentation at the server side. Hanano et al [15] and Kang et al [18] propose a hybrid network architecture consisting of cellular and Wi-Fi networks. The papers describe a file exchange algorithm in a peer to peer network with the help of a central server. Another proposed solution by Dorial et al [11] to deal with P2P downloads is by using multicast groups. A single node is responsible for distributing a given range of file segments to mobile nodes. The file is not striped across all mobile nodes in this proposed approach. Huguenin et al [17] propose an approach for peer to peer video on demand. The file segments are downloaded in such a way that the less advanced mobile devices get the latest file segments from the server which is sent to the most advanced peer. In return, the least advanced peer downloads the file segments from the most advanced peer in order. This approach ensures that all mobile devices take part in the file download process and ensure that deadlines are met in a timely fashion.

The concept of file segmentation and P2P download has been implemented since the start of Napster, Gnutella and BitTorrent. However, since we are using mobile devices that bring into consideration the challenges of mobility, resource constraints and time, the implementation of BitTorrent directly on mobile devices may not be practical. Ekler et al [12] present a comparison study by implementing the BitTorrent application on Nokia's Symbian platform and the Java Micro Edition platform. The aim of the authors was to examine the realistic possibility of constructing a peer to peer network using mobile phones and investigating the challenges faced. Though Sangam uses the concept of file segments and peer to peer networks, the idea of chunk assignments based on resources of a group of devices and multiple radio interfaces is not mentioned by authors.Claudino et al [10] implemented a file sharing application over Wi-Fi interface on the version 1.6 of the Android platform and used an existing Bluetooth file sharing application. The authors conclude that Wi-Fi is the preferred interface to transfer files in a peer to peer network. These evaluations allow us to compare our results with [10] over the Wi-Fi interface.

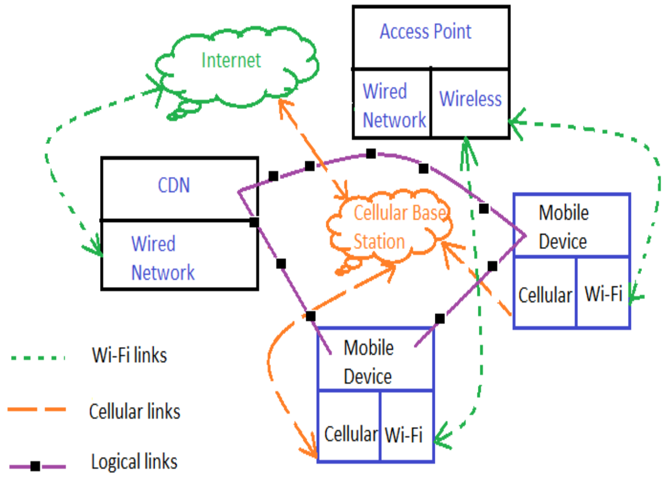


Figure 2: Network Model

Tysowski et al [24] implement a file sharing protocol on the Blackberry and Windows Mobile. The mobile nodes download a fraction of the file from the server using the cellular connection and then distribute file segments among themselves using the Bluetooth interface. The results show decreased download time and throughput for a large number of peers. Another implementation is seen in the Push and Track system by Whitbeck et al [27] for disseminating content to a large number of mobile node with a defined time deadline. The system makes use of the adhoc connections in the peer to peer network and reduces the load on the wireless infrastructure. The push and track system does not deal with large video files but is more tuned towards traffic updates and news.

3. MODELS AND ASSUMPTIONS

Network Model: The network model as seen in Figure 2 assumes a group of co-located mobile devices that are equipped with cellular and Wi-Fi interfaces. We call this a *group of peers* denoted as $V = 1, 2...., N$ with group size $|V| = N$. The mobile phones are connected via P2P to each other over the Wi-Fi interface, and use the Wi-Fi access point(AP) to communicate with each other. [1] CDN connections between all group peers $\{1, 2, .., N\}$ and the server S will be over the cellular network. The communication connections $\{e_1, e_2, ..., e_p\}$ among the group peer devices and the server S represent edges $E = \{e_1, e_2, .., e_p\}$ of the network group $G = (V, E)$ with $V = 1, 2...., N$ and $E = \{e_1, e_2, .., e_p\}$.

[1] Using adhoc mode to communicate among the group of peers will be carried out as an extension of our work.

At this point we assume G represents the group of co-located peer devices that are interested in downloading and sharing a common file. The Sangam system is represented by $Sangam_{sys} = G \cup \{S, \{(S,1), ..(S,N)\}\}$

Data Model: Let the file of interest F be divided into m chunks represented by $\{f_1, f_2 ..., f_m\}$. Each of these chunks resides on atleast one peer once the chunk assignment and download from the CDN is complete.

Resource Model: The battery level and CPU speed of the device queried and stored at the server S to assist with the chunk assignment and scheduling. The server assigns chunks based on all resources of the entire group of peers.

Mobility and Access Model: We assume that the nodes of the group are static or are moving very slowly, ensuring that all the peers in the group co-located.The model assumes that group of peers requests a common file of interest (F) from the server S within a registration time interval $t_arrival$ which is approximately a few seconds long. It means, within our design, we currently assume to do chunk assignment and scheduling to peers for a static group of peers $G' \subset G$ with $V' = \{1, 2, ..., n\} \subseteq V = 1, 2, .., N$ who register with the server S within an interval $t_arrival$. After this interval $t_arrival$, a new request to join the group of peers and download a file F is treated as a new group request where server S groups all new requests from new group of peers $G'' \subset G$ and derive new chunk assignments and schedule. The scheduled chunk assignments are derived and disseminated to all peers in the group of peers once atleast two peers have registered with the server S.

4. PROBLEM DESCRIPTION

We know that the maximum data rate that we can achieve using a cellular interface is less than what we can get using a Wi-Fi interface. When multiple mobile nodes request the same file using multiple interfaces, co-located group of peers can download the same file in a reduced amount of time. The total number of chunks m in a given file is $m = \frac{FileSize}{ChunkSize}$. The number of chunks assigned by the server S to a peer i is m_i. m is the total number of chunks in the file of interest F and is given by $\frac{m}{N}$. The total number of peers is represented by N. Let T_{switch} be the time required to switch interfaces from cellular to Wi-Fi. Let $t_{ij}(f_x)$ be the time to download a chunk f_x from source i to destination j. The source i can be a server or a mobile device i but the destination will always be a mobile device j, j in V. We obtain the best chunk assignment with optimal file download time subject to constraints such as cellular bandwidth, Wi-Fi bandwidth, number of chunks, size of group of peers and the gain metric. We define our gain metric as follows:

$$Gain = \frac{T_{hybrid}}{T_{cell}} \leq 1 \tag{1}$$

We will validate the gain metric through our experimental results shown in Section 7. The overall goal of the group file sharing, is to maximize under constraints such as bandwidth availability for group of peers in cellular and Wi-Fi networks, file size, CPU and power of the device. We use these constraints to formulate the download time optimization problem. The problem is NP complete because the download time optimization problem includes derivation of the optimal chunk assignment and schedule [25] [9].Hence, we need heuristic algorithms to solve the optimization problem.

To download the assigned chunks m_j from the server the total time for any peer in a group of peers is given by:

$$\sum_{k=1}^{m_j} t_{sj}\left(f_k\right) \tag{2}$$

The total time taken by peer i to download the remaining chunks from a group of peers is given by:

$$\sum_{j=1, j\neq i}^{n} \sum_{k=1}^{m_j} t_{ji}\left(f_k\right) \tag{3}$$

The optimization problem for minimizing the download time of a file F to a group G of n peers:

$$\begin{aligned} \textbf{min } T(F)_{\{i,j,k\}} = \sum_{i=1}^{n}\sum_{k=1}^{m_i} t_{si}\left(f_k\right) + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\sum_{k=1}^{m_j} t_{ji}\left(f_k\right) \\ + n\,T_{switch} \ \ \textbf{subject to} \end{aligned} \tag{4}$$

$$B_{cell} \leq B_{cell}^{max}$$

$$B_{wifi} \leq B_{wifi}^{max}$$

$$m_i \leq m$$

$$gain \leq 1$$

$$n \leq N$$

$$E_i \leq 100$$

5. SANGAM ARCHITECTURE

The design of Sangam architecture encompasses registration of the group of peers at the server S, chunk assignment and schedule by the server based on the resources at each peers of the group, and the download protocols carried out at each of the peers. Each peer

Table 1: Notations

Symbol	Description
m	Total number of chunks in the file of interest
m_i	chunk assigned to peer i
n	Size of registered Group of Peers, $n \leq N$
N	Maximum Size of Group of Peers
E_i	Battery level of peer i
$t_{ij}\left(f_x\right)$	Download Time of chunk f_x from source i to destination j
T_{cell}	Download time of entire file over cellular interface
T_{hybrid}	Download time of entire file over hybrid network
T_{switch}	Time taken to switch between the cellular and Wi-Fi interface
B_{cell}^{max}	Maximum cellular bandwidth achieved by peer i
B_{wifi}^{max}	Maximum Wi-Fi bandwidth achieved by peer i
B_{cell}	cellular bandwidth achieved by peer i
B_{wifi}	Wi-Fi bandwidth achieved by peer i

has to register with the server S that contains the file of interest. The registration process involves the exchange of information between the server and group of peers. The server performs coordinated chunk assignment on the file of interest based on the received resource information from each peer. Finally each peer downloads the assigned chunks from the server and group of peers. The overall Sangam flow is as follows:

1. A peer in the G group registers with the server S, specifies the file of interest F within interval $t_arrival$.

2. Each peer during the registration phase also declares resource information such as CPU speed, level of battery power and available bandwidth to the server S.

3. The server S performs coordinated chunk assignment for G' group of registered peers on the file F, based on the received resource information from group of registered peers in the interval $t_arrival$.

4. The server S schedules the download of assigned chunks from the server over cellular-CDN network to the group of registered peers.

5. The peers of the group exchange missing chunks among each other via Wi-Fi - P2P network.

5.1 Coordinated Chunk Policies

The servers upon completing the registration process for group of peers G'
inG begins assigning the chunks to the different peers. The common file of interest F is divided into m chunks. Each peer is assigned a range of chunks based on different policies as a result of the heuristic approaches to solve the optimization problem in (4). The peers are then sent the chunk assignments along with the G' group of peers information *peer list* at the end of the registration process. The cellular connection between the server S and the G' group of peers is similar to a control channel. The server is aware of the chunk assignments and the group of peers are informed of this to assist in the download process.

We propose three different approaches to chunk assignment for the registered group of peers. We present (1)equal chunk assignment,(2)chunk assignment based on battery level, and (3)chunk assignment based on CPU speed.

1. *Equal Chunk Assignment - Homogeneous system:*

   In the simplest chunk allocation algorithm, all peers are assumed to be equal as shows in Figure 3. This approach serves as the baseline against which we compare other approaches. The server S calculates the total number of chunks and divides them equally among the co-located group of peers. In this chunk assignment policy, the server S does not consider the heterogeneity of the G' group of peers requesting the common file of interest F. Peers are assumed to have similar battery levels and processor speeds. In essence, the server assumes that they are capable of carrying out the download process at the same capacity as one another. This chunk assignment is what we would expect in a homogeneous environment where all mobile phones not only have the same underlying hardware but also the same version of the operating system and radio interface optimizations. In the following equations, R_i^{equal} is the fraction of chunks assigned to each peer, P_i^{equal} is the number of chunks assigned to each peer and m is the total number of chunks.

$$R_i^{equal} = \frac{1}{n} \tag{5}$$

Table 2: Notations

Symbol	Description
m	Total number of chunks in the file of interest
n	Size of registered group of peers
R_i^{equal}	Fraction of the file of interest assigned to peer i
P_i^{equal}	Number of chunks of the file of interest assigned to peer i
$R_i^{battery}$	Fraction of the file of interest assigned to peer i based on its battery level
$P_i^{battery}$	Number of chunks of the file of interest assigned to peer i based on its battery level
R_i^{cpu}	Fraction of the file of interest assigned to peer i based on its CPU speed
P_i^{cpu}	Number of chunks of the file of interest assigned to peer i based on its CPU speed
W_i	Weighted Ratio of the battery level and CPU speed of peer i

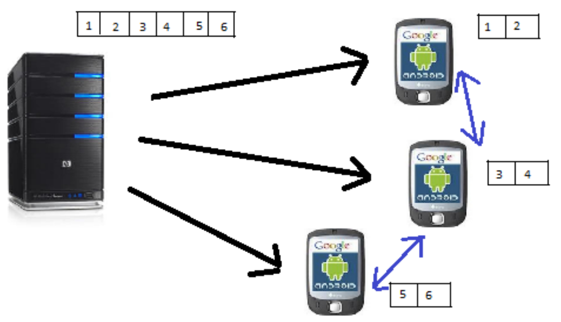


Figure 3: Equal Chunk Assignment - Homogeneous System

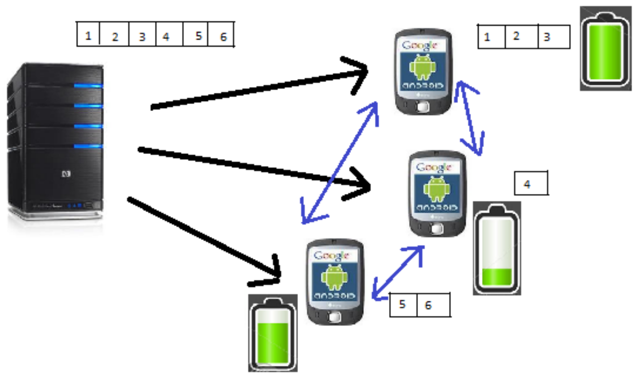


Figure 4: Chunk Assignment based on Battery Level - Heterogeneous system

Chunk assignment:

$$P_i^{equal} = R_i m = \frac{m}{n} \tag{6}$$

Note that the chunks are scheduled at S in the FCFS registration order.

2. *Chunk Assignment based on Battery Level - Heterogeneous system:*

   For any mobile phone to carry out a file download protocol using a cellular or a Wi-Fi link or both, the battery power required for transmission is one of the highest cost compared to other processes running on the mobile device as shown in Figure 4. For a mobile node j whose battery power is relatively lower than the power of other mobile nodes i in the group G', it might be beneficial to conserve the energy and be able to view the video instead of completely draining its battery power. Another reason for other nodes i to support this unequal chunk management is to ensure that all file chunks are downloaded and present in the G' group of peers. If a mobile device j loses its battery power in the middle of the file download, all peers will be forced to request all chunks assigned to the peer j from the server S over their cellular interface. This defeats the purpose of the cellular- Wi-Fi CDN-P2P protocol. When each of the peers using this modified chunk assignment registers with the server S, they send the server S their current battery level. The server proceeds to calculate the ratio of total chunks that the peer can download. Here $R_i^{battery}$ is the fraction of chunks assigned to each peer i, E_i is the battery level of peer i and $P_i^{battery}$ is the number of chunks assigned to peer i.

$$R_i^{battery} = \frac{E_i}{\sum_{k=1}^{n} E_k} \tag{7}$$

Chunk assignment:

$$P_i^{battery} = R_i^{battery} m \tag{8}$$

Note that all chunks are scheduled at the server S in the FCFS registration order.

3. *Chunk Assignment based on CPU Speed - Heterogeneous system:*

   This algorithm takes into account a more realistic setting in the current mobile ecosystem. With the

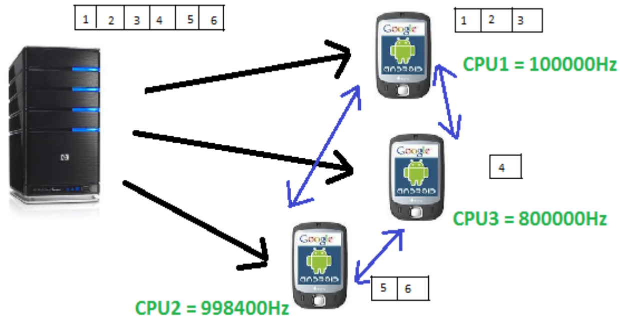


Figure 5: Chunk Assignment based on CPU Speed - Heterogeneous system

increased rate of mobile development, each mobile node can be composed of different processors and hence can work at different speeds as shown in Figure 5. In order to account for this heterogeneity, each peer sends its maximum CPU speed (e.g. 800MHz, 1GHz) to the server during the registration process. The server calculates the ratio that each of the peers needs to download based on a similar calculation as shown above. A faster processor might be handed a larger chunk assignment from the server S as per this algorithm. Though there is an uneven chunk assignment among the peers, there is a benefit for faster mobile nodes to download a larger number of chunks from the server. When there is equal chunk assignment we notice that the slower peers become a bottleneck for completing the download of the file. The faster peers either wait for the slower peer to complete the file download of assigned chunks or end up requesting the chunks from the server directly. The cellular bandwidth may be the bottleneck for performance of faster peers when downloading chunks from the server. However, peers with faster CPU speeds perform at a very high rate when exchanging chunks within the group of peers. Hence, the benefits of faster peers can be utilized by using CPU speed for chunk assignment. Here R_i^{cpu} is the fraction of chunks assigned to each peer i, S_i is the CPU speed of the peer i and P_i^{cpu} is the number of chunks assigned to peer i.

$$R_i^{cpu} = \frac{S_i}{\sum_{k=1}^{n} S_k} \tag{9}$$

Chunk assignment:

$$P_i^{cpu} = R_i m \tag{10}$$

Note that all chunks are scheduled at the server S in the FCFS registration order.

5.2 Sangam File Download Protocol

The download protocol takes place once the peer registration and chunk assignment are completed. In order to download the entire file F, the peer has to go through four download phases.

In the first phase, each peer i requests the server S for the chunks assigned to it in the FCFS registration order. Between the first and seconds phase the communication interface is changed from cellular to Wi-Fi. The mobile devices today have their default interface set to Wi-Fi interface. As mentioned previously, we can direct traffic over the cellular interface for a particular destination. However, the cellular interface eventually disconnects leaving only the Wi-Fi interface. We suspect that both interfaces are not allowed to be switched on at the same time due to an underlying system policy. However, the recent release of the Android 4.0 is able to support both cellular and Wi-Fi traffic using the *WiFidirect* package. The simultaneous turning on of multiple interfaces might also prove to be too much of a drain on the battery life since these devices have not been optimized to carry out network connections on two different interfaces.

In the second phase, each peer i requests the G' group of peers for the remaining chunks as per peer list and chunk assignment handed down from the server at the end of the registration process. In order to avoid overloading any one peer in a given time interval by requesting the same set of chunks, we use a round robin scheduling policy locally at each peer. Each peer begins phase two of the download process by requesting the peer whose ID is one greater than its own ID. The peer ID is issued by the server S at the time of registration.

The third and fourth phases take place only if a peer is unable to download all chunks from the server S and G' group of peers in the first two phases. In the third phase, a mobile node requests the peer for the chunk that it was unable to download in phase two. There is an interval of time during which the mobile node might be unreachable due to switching between cellular and Wi-Fi interfaces. Since we assign chunks based on the capability of the nodes, the mobile nodes will not complete phase one at the same time. The interface switch time T_{switch} varies among devices with an average value of 12s in our experiments. A few chunks of the file that might not be obtained from each of the peers at the beginning of phase two is acquired by the peer in phase three.

The fourth phase takes place if the chunk is not present among the group of peers. The mobile node then requests the server for any chunks that might not have been downloaded by the peers. If the per node chunk

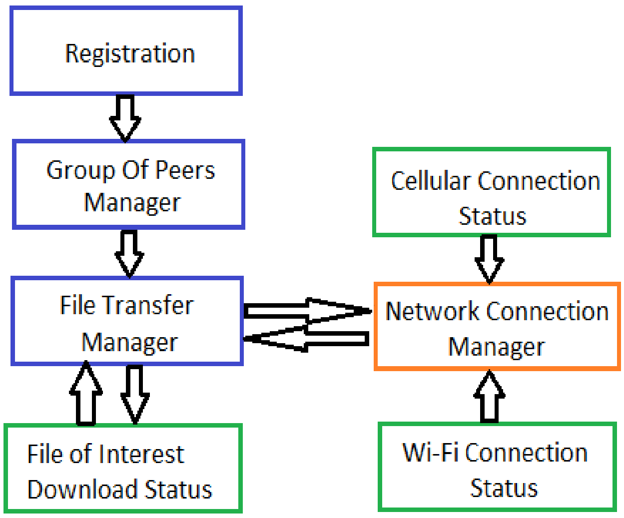


Figure 6: Sangam Client Architecture

assignment is small, there is a chance that the faster nodes are not able to get all the chunks from the other peers in a timely fashion and end up requesting a larger portion of the remaining chunks from the server in the end.

6. IMPLEMENTATION

Sangam has been implemented on the Android operating system version 2.1 and above. The file download application on the client side is present in the userspace and uses the Java API for socket connections. The turning off and on of the cellular and Wi-Fi interface is carried out using the *WifiManager* class of the Android API. The phones used in the experimentation are Droid, Nexus S and Nexus One. In figure 6 we highlight the main modules involved in the implementation of the client Sangam framework. The server consists of a simple C program that accepts client connections and responds with the request file chunk. The server handles the registration of the group of peers and the chunk assignment based on the received resources from the group of peers.

Registration: Registration of peer at the server S allows the user to specify the file of interest F, IP address and port number.The registration process also involves the mobile node sending information about the battery level and CPU speed to the server. In order to ensure that we are able to set up TCP connections over the cellular and Wi-Fi interface, each mobile node has to send the IP address of the Wi-Fi interface along with the IP of the cellular interface. There could be a number of ways to accomplish this. The mobile node could have a static IP address associated with its Wi-Fi interface. In our current implementation we turn on the Wi-Fi interface and record the IP address. The Wi-Fi interface is then turned off to allow for the cellular interface to come up. The android phone by default chooses the Wi-Fi interface to connect to and hence an explicit call using the *WifiManager* is used to turn off and turn on the interface. Each peer is also assigned an ID that is used in the download process. As we have previously stated in our model, we assume that the server waits for the minimum number of peers to register in a certain time interval $t_{arrival}$. If the minimum number of peers does not register within $t_{arrival}$, then each peer receives the entire file over the cellular link and no sharing policies are implemented by the server S.

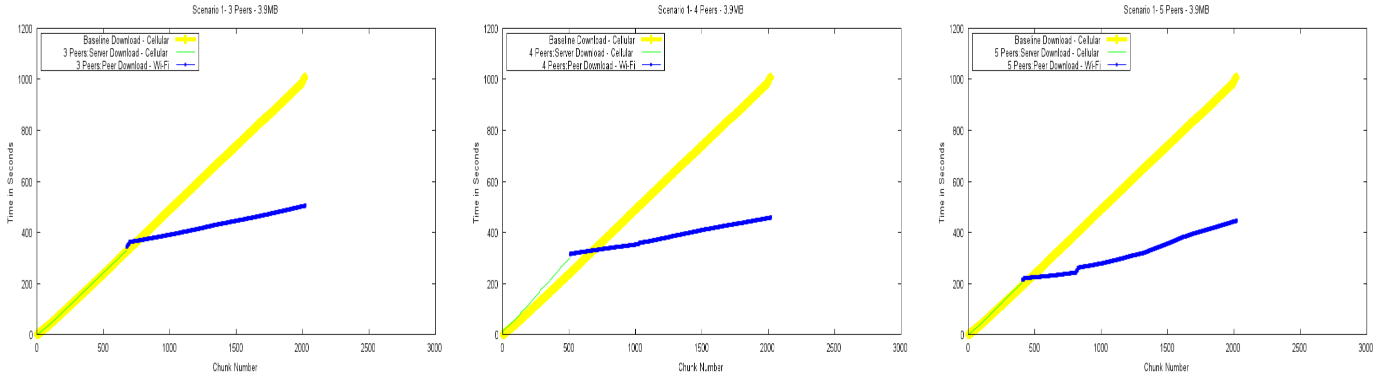
Group of Peers Manager: After registration with the server S, the mobile node receives the list of peers interested in the same file F along with their IP address and port information. Peer management handles the storage of the IP address, port number, and the chunks $\{f_1, f_2..., f_k\}$ assigned to each of peers who are interested in the same file F. The peer list includes an array of peers in G' and their chunk assignment $\{f_1, f_2..., f_k\}$. The peer list is used to request the chunks from the group of peers in phase 2 and phase 3 of the download process.

Network Connections Manager: There are three network connections that the peer handles during the file download process. The first two are used request file chunks from the server and group of peers. The third one is for handling incoming file chunk requests from other peers. The connection to the server takes place over the cellular network. The other two peer connections take place simultaneously over the Wi-Fi interface. The current cellular and Wi-Fi connection status is maintained by the network connections manager. We use TCP protocol to carry out all file requests.

File Transfer Manager: The file transfer manager is responsible for carrying out the four phases of the download protocol. It is responsible for switching interfaces between cellular and Wi-Fi in a timely fashion. It keeps track of the number of chunks received from the group of peers and services request according to the chunk assignment handed down from the server. It also ensures mutual exclusion when writing and reading chunks from a given output file. The video file is present at the server S and divided into chunks of size 2048 bytes. We carried out file download evaluations with chunk sizes varying from 256bytes to 4096bytes. We found 2048bytes to be the optimal size which balances the limited memory requirements of the mobile device and the IP packet size. The video files used are 2.2MB and 3.9MB.

7. EVALUATION

The evaluation of the Sangam framework is carried



(a) (b) (c)

Figure 8: Scenario 1: File Download Time with Equal Chunk Assignment Approach

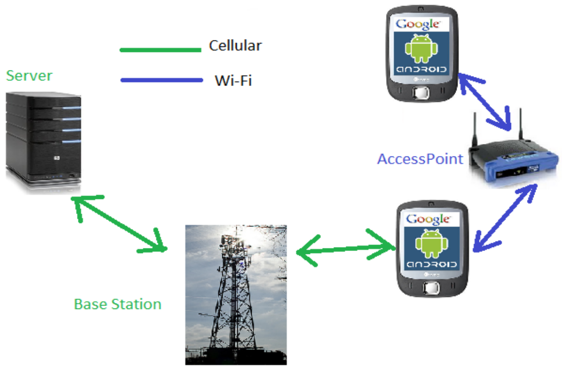


Figure 7: Experimental Setup

out on Android phones having an operating system version of 2.1 or greater. We use five Droid phones which use the Verizon network for cellular connectivity. To demonstrate the heterogeneity of devices we use the Nexus One device with a T-mobile cellular connection. The Verizon network uses CDMA technology and offers an average speed of 848kbps [4]. T-mobile uses GSM technology with an average speed of 7.2Mbps [1]. A maximum of five phones were used during this evaluation and files of size 2.2MB and 3.9MB were requested. We use 802.11x for our Wi-Fi connectivity. The network uses WPA2 protocol for authentication.

In our evaluation, the **file download time** $T(F)$ is the most important metric. Our goal is to minimize the time required by all peers to download the complete file. In some cases, when we assign chunks based on battery level, our goal is also to have as low battery consumption of certain peers as possible. Each resulting plot has the baseline cellular download time, server (CDN) download time over the cellular network and peer(P2P) download time over the Wi-Fi interface. The *Baseline Download - Cellular* is the time taken to download the entire file from the server over the cellular network(CDN download). The *Server Download - Cellular* is the time taken by a peer to download the assigned chunks from the server over the cellular link and *Peer Download Wi-Fi* is the time taken to download the remaining chunks from the group of peers over the Wi-Fi interface (P2P download). We have conducted Baseline measuments where 5 peers simultaneously request the entire file of size 2.2MB from the server S over the cellular network. The average download time was found to be 551.375 seconds with a standard deviation of 38s over three trials. The download time per file chunk of size 2048 bytes was 0.494 seconds.

Our original aim was to build the system with simultaneous download over both cellular and Wi-Fi interface of the mobile device. However in the real system framework we ran into many problems. For example, using the requestRouteToHost function in the Connectivity Manager class of the Android SDK [3] we were able to gain marginal success by specifying the interface to be used for a particular IP. However, it was difficult to sustain the connectivity of the cellular connection since the Wi-Fi interface is the default network to be used. The increased drain of the battery power and possible interference between the two transmissions was one of the reasons for us to switch to a sequential download process as opposed to a concurrent one. We envision a change in the hardware of mobile devices which will allow us to use simultaneous network interfaces similar to the design mentioned it [21]. The addition of the WifiP2pManager class to the Android SDK starting from Android 4.0 will enable us to use the Wi-Fi interface for peer to peer ad hoc connections while simultaneously maintaining our cellular connectivity [5].

7.1 Scenario 1 - Equal Chunk Assignment

In our first set of experiments we demonstrate the advantage of the hybrid network using the Sangam framework. We assign file chunks equally to all the peers in the group G'. The group of peers G' request a file F of size 3.9MB from the server S. We carry out this request for different group sizes 3, 4, and 5 registered peers. In Figure 8, the download graphs (8(a), 8(b), 8(c)) show that as the number of peers increases from 3 to 5, the download time of one peer decreases by 60s. Each peer is downloading a smaller number of file segments over the cellular interface. The majority of the chunks are now downloaded from the group of peers over the Wi-Fi interface. There is a decrease in the gain ratio due to the decrease in download time over the cellular interface.

In Figure 9 we observe that the download time may not always give us the best gain ratio. In the Figure 9 plots, we consider a peer-to-peer network of size 3 peers who request 2.2MB from the server S. In both plots we see that the time to complete the file download is only 10s to 50s lesser than the baseline case. We attribute this to two factors:

Interface Switch Time: After the completion of phase 1 of the download process over the cellular network, all peers switch their interface to Wi-Fi. The average time overhead for the switching of interfaces from cellular to Wi-Fi is 10s to 15s on average. However, some peers might not be able to scan and associate with the Wi-Fi access point within this interval of time. The longer the mobile device takes to associate and connect to the Wi-Fi access point, the greater the time is that the node is unreachable by all other peers. Hence peers end up waiting for any device that is unreachable.

Contention: Multiple devices in the peer to peer network contend with one another in order to be scheduled by the Wi-Fi access point. This takes place when the number of access points are few in number. A contention graph is formed and results in severe degradation in the performance of the hybrid network. Multiple peers end up waiting to download from their neighboring peers. We observe the file download process taking place in a subset of peers in a given interval of time. The gain that can be achieved in this situation is very low.

7.2 Scenario 2 - Chunk Assignment based on Battery Level

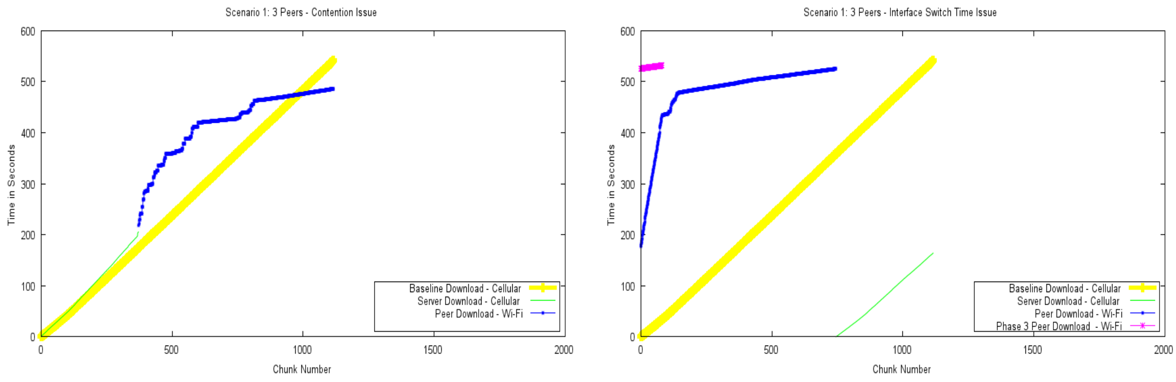
In our next experimental scenario we consider chunk assignment based on the battery level. The peer with the greater battery level is assigned a larger share of the file segments. In this case the metric that we choose to minimize the file download time under the constraints of the battery consumption of the peers. We consider a hybrid network with 3 and 5 peers requesting a file of size 2.2MB. In Figure 10, we show the result of the hybrid network of 3 and 5 peers, and the download time $T(F)$ is only slightly greater than the scenario of equal chunk assignment. Table 3 also shows that the peers with a lower battery level seem to suffer a greater decrease in battery power than the peers with a higher battery level. Hence, having such peers take on a greater a share of the cellular download would completely exhaust the remaining power. When the number of peers is 3, we see in Figure 10(a) that the peer with a lower battery level ends up waiting for the cellular download to be completed. We also notice in Table 3 that the time to download the chunks from the group of peers for a peer with battery level 20 is larger than it is with battery level 50 and 70. Due to the limited time spent downloading the chunks over the cellular interface, the peer does not suffer from a large drop in power level.

Peer ID	Initial Battery Level	Final Battery Level	Download time(s)
0	50	50	251.351
1	40	20	238.686
2	80	80	285.429
3	40	20	288.402
4	30	30	241.465

Table 3: Scenario 2: Download Time for 5 Peers

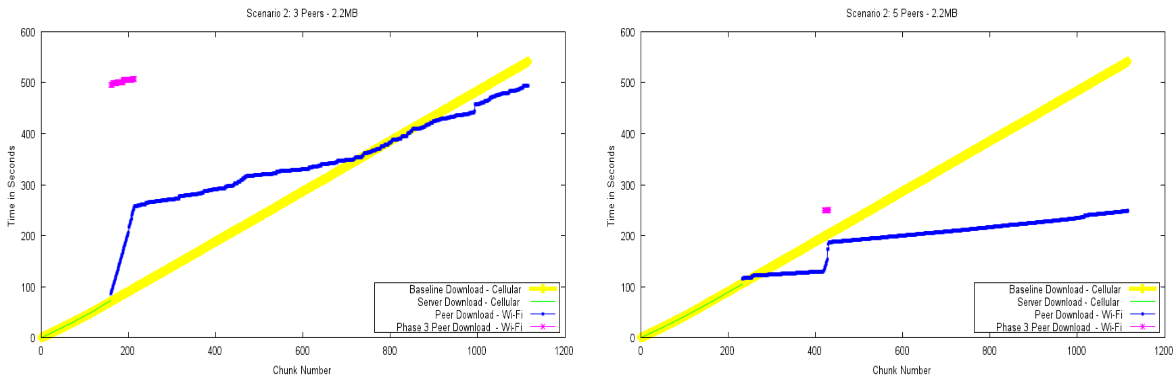
7.3 Scenario 3 - Chunk Assignment based on CPU Speed

In order to account for the heterogeneity of nodes we assign chunks to the group of peers based on the CPU speed level of the mobile devices. We use two Droids that use the Verizon cellular network and one Nexus one that uses the T-Mobile cellular network. The CPU level of the Droid phones are 800000Hz and that of the Nexus One is 998400Hz. All 3 peers request a file of size 1MB from the server. The Nexus One is assigned a greater share of the file segments due to the higher CPU speed. In Figure 11(a), the initial download from one of the Droids carried out by Nexus One is slightly slower in the beginning. This may be because of the interface switch that is taking place at the corresponding Droid. The plot of the corresponding Droid shows a certain delay in switching interfaces and starting the download from the group of peers. We notice the speed of download of the remaining chunks from the group of peers by the Nexus One device is extremely fast (see Figure10(a)). The Droid is able to leverage the faster Nexus One device in order to complete its download. However, the faster CPU power of the Nexus one was not able to overcome the bottleneck of the throughput of the cellular network. As we can see on the plot in Figure10(b) of the Nexus



(a) (b)

Figure 9: Scenario 1 : Impact of Contention and Interface Switch Time on File Download Time

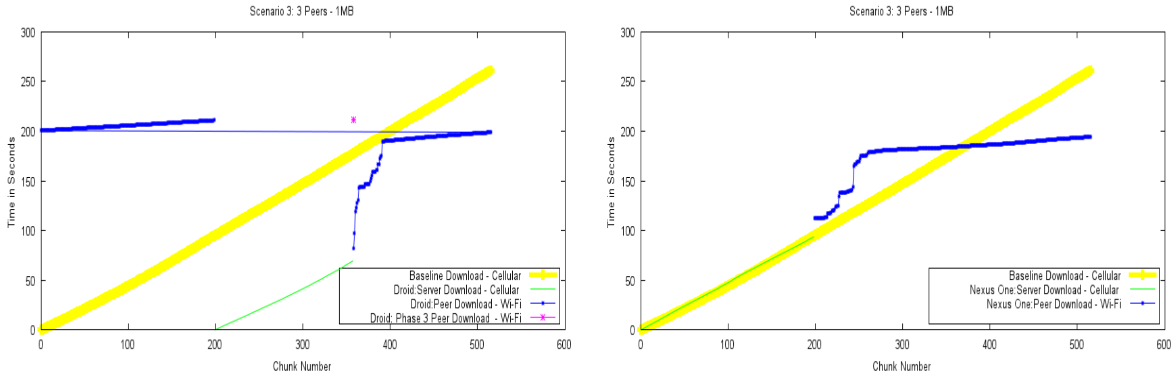


(a) (b)

Figure 10: Scenario 2 : File Download Time for Chunk Assignment Approach based on Battery Level

Device/Model	Network Interface	Network Model	Network Protocol	Number of Peers	File Size(MB)	Download Time(s)	Evaluation Framework
Blackberry [24]	Bluetooth	P2P	OBEX	2	3	4	Implementation
Windows Mobile [24]	Bluetooth	P2P	UDP	3	9.15	300	Implementation
Nokia N70 [20]	Cellular, Bluetooth	Hybrid	TCP	2	-	$\frac{T_{cellular}}{2}$	Implementation
Ishare [26]	Cellular, Wi-Fi	Hybrid	TCP	15	3.9	500	Simulation
Sangam	Cellular, Wi-Fi	Hybrid	TCP	5	3.9	442.106	Implementation

Table 4: Comparison of File Sharing Techniques



(*a*) (*b*)

Figure 11: Scenario 3 : File Download Time for Chunk Assignment Approach based on CPU Speed: (a)Droid ,(b)NexusOne

One mobile device, the cellular download line mimics the baseline plot inspite of having a greater processing power.

7.4 Comparison with Ishare

As previously mentioned in Section 3, Ishare protocol [26] carries out file download from the server and peers simultaneously and it is validated only via simulation. We compare our implementation to the simulation results presented in Vu et al [26]. The simulation was carried out using Network Simulator 2 (NS2) with segment size 4KB. The simulator uses IEEE 802.11b adhoc link and 1xEV-DO cellular link with a peak rate of 2.4Mbps. We only compare Scenario 1- equal chunk assignment with the plot displaying average download time. The Ishare evaluation uses the Random way point mobility model as opposed to static or slow moving nodes in the Sangam protocol. We notice that the download time in the Ishare simulation is 500s which is higher than our implementation. This might be because the group size of Ishare is 15 which is much larger than 5. Futhermore the cellular rates assumed and mobility models affect the file download time. Note that we do not assume mobility in the Sangam implementation due already major implementation challenges such as heterogeneity, variety in battery power levels and overhead in switching between multiple radio interfaces, of file sharing in static topology scenarios. We have identified these practical challenges and developed feasible solutions.

8. CONCLUSION

The increased use of mobile devices have led to the congestion of cellular networks. Recognizing that most mobile devices are equipped with multiple interfaces, we propose to develop a solution that would involve downloading a file over both the cellular and Wi-Fi links. In scenarios where there is no infrastructure and co-located peers are interested in fast download of large files, the proposed Sangam framework minimizes the file download time. Using efficient chunk management policies based on resource information from the group of peers, the file chunks are assigned and distributed accordingly among the group of peers.

We implement the Sangam framework on Android phones in order to compare and contrast the three chunk management policies - 1) equal chunk assignment 2)chunk assignment based on battery level and 3) chunk assignment based on CPU speed. We obtain a decrease in download time of 50s -60s by implementing equal chunk assignment and CPU speed approach. Though the download time is slightly higher when we implement chunk assignment based on battery level, we are able to conserve the power level of certain mobile devices ensuring the completion of the download process.

9. ACKNOWLEDGMENTS

This work was funded by the Boeing - ITI(Information Trust Institute) center.

10. REFERENCES

[1] Cdma and gsm network speed. http://www.tested.com/news/cdma-vs-gsm-examined-which-3g-network-is-superior 1630.
[2] Cisco newsroom. http://newsroom.cisco.com/press-release-content?type=webcontent.
[3] Connectivity manager android api. http://developer.android.com/reference/android/net/ConnectivityManager.html.
[4] Speedtest data. http://www.wired.com/gadgetlab/2011/02/iphone-speedtest/.
[5] Wi-fi direct api. http://developer.android.com/resources/

samples/WiFiDirectDemo/index.html.

[6] Efficient data transfer over cellular networks, 2006-2009.

[7] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 280–293. ACM, 2009.

[8] R. Bhatia, L.E. Li, H. Luo, and R. Ramjee. Icam: Integrated cellular and ad hoc multicast. *IEEE Transactions on Mobile Computing*, pages 1004–1015, 2006.

[9] G.C. Buttazzo. Basic concepts. *Hard Real-Time Computing Systems*, pages 23–51, 2011.

[10] J. Claudino, N. Shay, and D. Valenzona. An evaluation of peer-to-peer file sharing on the android mobile platform.

[11] S.S. Doria and M.A. Spohn. A multicast approach for peer-to-peer content distribution in mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6. IEEE, 2009.

[12] P. Ekler, J.K. Nurminen, and A. Kiss. Experiences of implementing bittorrent on java me platform. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 1154–1158. IEEE.

[13] R. Gass and C. Diot. An experimental performance comparison of 3g and wi-fi. In *Passive and Active Measurement*, pages 71–80. Springer, 2010.

[14] D. Gómez-Barquero, A. Fernández-Aguilella, and N. Cardona. Multicast delivery of file download services in evolved 3g mobile networks with hsdpa and mbms. *Broadcasting, IEEE Transactions on*, 55(4):742–751, 2009.

[15] H. Hanano, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito. Video ads dissemination through wifi-cellular hybrid networks. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[16] H.Y. Hsieh and R. Sivakumar. On using peer-to-peer communication in cellular wireless data networks. *Mobile Computing, IEEE Transactions on*, 3(1):57–72, 2004.

[17] K. Huguenin, A.M. Kermarrec, V. Rai, and M. Van Steen. Designing a tit-for-tat based peer-to-peer video-on-demand system. In *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, pages 93–98. ACM, 2010.

[18] S.S. Kang and M. Mutka. Efficient mobile access to internet data via a wireless peer-to-peer network. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 197–205. IEEE, 2004.

[19] H. Luo, X. Meng, R. Ramjee, P. Sinha, and L.E. Li. The design and evaluation of unified cellular and ad hoc networks. *IEEE Transactions on Mobile Computing*, pages 1060–1074, 2007.

[20] T.K. Madsen, Q. Zhang, F. Fitzek, and M. Katz. Exploiting cooperation for performance enhancement and high data rates. *Journal of Communications*, 4(3):193–202, 2009.

[21] Y.T. Man and QI Yihong. Mobile wireless communications device having dual antenna system for cellular and wifi, March 5 2008. EP Patent 1,895,383.

[22] G.P. Perrucci, F.H.P. Fitzek, G. Sasso, W. Kellerer, and J. Widmer. On the impact of 2g and 3g network usage for mobile phones' battery life. In *Wireless Conference, 2009. EW 2009. European*, pages 255–259. IEEE, 2009.

[23] I. Rimac, S. Borst, and A. Walid. Peer-assisted content distribution networks: performance gains and server capacity savings. *Bell Labs Technical Journal*, 13(3):59–69, 2008.

[24] PK Tysowski, P. Zhao, and K. Naik. Peer to peer content sharing on ad hoc networks of smartphones. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 1445–1450. IEEE.

[25] J.D. Ullman. Np-complete scheduling problems*. *Journal of Computer and System Sciences*, 10(3):384–393, 1975.

[26] L. Vu, K. Nahrstedt, I. Rimac, V. Hilt, and M. Hofmann. ishare: exploiting opportunistic ad hoc connections for improving data download of cellular users. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 1475–1480. IEEE, 2010.

[27] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M.D. De Amorim. Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays. *Arxiv preprint arXiv:1007.5459*, 2010.