

INDAPSON: An Incentive Data Plan Sharing System Based on Self-Organizing Network

Tuo Yu*, Zilong Zhou*, Da Zhang*, Xinbing Wang*, Yunxin Liu†, Songwu Lu§

*Dept. of Electronic Engineering, Shanghai Jiao Tong University, China

†Microsoft Research Asia, Beijing, China

§Dept. of Computer Science, University of California, Los Angeles

Email: {yutuo, zhouzilong, johnson92104, xwang8}@sjtu.edu.cn, yunxin.liu@microsoft.com, slu@cs.ucla.edu

Abstract—The contradiction between dynamic user traffic and fixed data plans has drawn increasing attention in the field of mobile applications. In this paper we build a data plan sharing system named INDAPSON to consider a scenario where some smartphone users have surplus data traffic and are willing to help others download data. Virtual credits can be gained as reward, which can be used to ask for future help in downloading. To realize this model, we make the following contributions: 1) A dynamic self-organization strategy to adapt to mobile terminals; 2) An incentive mechanism named RAP to encourage participation; 3) Power-saving strategies to reduce power consumption. The main advantage of our system is that users gain improvement in download rate while being able to convert their surplus data traffic to virtual credits. The results of experiment and simulation show that users in our system can manage their surplus data plan more efficiently while a high-speed download rate can be achieved.

I. INTRODUCTION

With the rapid development of mobile computing, today's mobile terminals, such as smart phones and tablets, are becoming more and more powerful and can offer people with highly convenient Internet access. Using mobile terminals to get Internet services is becoming increasingly important in people's daily life. People can browse webs or download data anytime and anywhere through their cellular connections. Indeed, cellular data traffic is growing rapidly these years. However, we have to notice that there exist many drawbacks of today's mobile data service, and we are interested in the following two common problems:

Firstly, we notice a problem that many ISPs don't provide unlimited data plans, and overuse is of high cost. For users who prefer tiered data plans with a limited amount of data, their network usage is always dynamic while their data plan is fixed. Sometimes their data plans are overused but underused at other times. Thus users have to pay attention on their data usage.

Secondly, with the increasing demand of data download through cellular connections, downstream rate offered by current cellular networks is insufficient to meet users' demand. Furthermore, the data rate of cellular network fluctuates severely over time. As a result, people often fail to get high quality of Internet service.

Thus, it is of great significance to figure out an effective mechanism to make use of idle wireless interfaces and improve cellular bandwidth without unacceptable expense to update the infrastructures of ISPs.

In this paper, we propose a novel INcentive DATA Plan Sharing system based on Self-Organizing Network (*INDAPSON*) to tackle the two problems mentioned above. Our proposal is based on two observations: 1) some users still have under-used data plan while some other users overuse their data plan. 2) people often stay together in daily life - for instance family members in houses, co-workers in meeting rooms, passengers on buses etc. These observations provide opportunities for our solution. The two goals that our system aims to achieve are 1) meeting users' dynamic traffic requirements and 2) increasing the download data rate.

In order to discriminate different participators, we classify users into three types in our system:

- *Primary user*: user who needs to download data from the Internet.
- *Assistant user*: user who helps a download user download data from the Internet.
- *Detected user*: user who acts as a candidate for assistant user.

Our system fits a scenario where assistant or detected users have idle cellular links or are willing to deal with their surplus data traffic, and it works as follows: one primary user initiates the construction of a local wireless network, the members of which are called detected users. Several assistant users are chosen from the detected users to help the primary user download data from the Internet through their cellular connections, and then relay data segments arrived to the primary user through WiFi connections. Since the low downstream rate of cellular networks is the bottleneck of system performance, we run multiple threads on primary user to receive data from different sources (server and peers) simultaneously, avoiding block on single cellular connection. After a cooperation, records for the whole download process are uploaded to a management server on the Internet. With a pricing scheme called *Reputation Adaptive Pricing (RAP)* applied, a primary user can pay a certain amount of virtual credits to assistant users, which can be used to purchase the help from other users next time.

One important feature of INDAPSON is that users do not necessarily know the existence of each other or are interested in the same data. They simply keep our application running in the background of their smartphones all the time without other manipulations. Some of previous works have discussed on the cooperative download within a small user group, where

members in this group download the same target file at the same time. Thus our system has a wider applicable scenario.

To make the exchange of data possible among users, it is necessary to construct a local network. One important challenge we faced with is how to construct such local network. Since users do not necessarily know each other, our application in background must listen to the request for help all the day time. Thus the problem of power consumption would limit the usage of our system. Moreover, due to the mobility of users, the local network is unstable. To solve these problems we have designed a network consisting of two layers: Bluetooth-Management Network and WiFi-Transmission Network. In the previous layer Bluetooth is used to manage the peers, and to model the join or quit of users, we use “detected user” to denote a user who has joined the local network but has no download task to execute. In the latter layer, we use WiFi to achieve a higher transmission rate. Power-saving listening method is applied to reduce the power consumption during idle time, and a heuristic scanning method is used to save power consumed in the Bluetooth-Management Network.

The main contributions of our work are listed as follows:

- We propose a data plan sharing system which allows users to better manage their data plans and make it possible for unfamiliar users to cooperatively download data from the Internet. The possible data traffic waste due to fixed data plans can be avoided, and the download rate is also increased. To the best of our knowledge, it is the first system to explore the utilization of user’s idle cellular links and to deal with the problem of fixed data plan.
- A scheme named RAP is applied for incentive. An assistant user can actually transfer its data usage into virtual credits. In the next cooperation where it acts as a primary user, it is able to utilize the credits in its account to purchase download bandwidth as well as data traffic from other users.
- Methods based on intermittent sleeping and heuristic strategy are used to save power during the construction of local network.
- Our system has been implemented on Android testbeds, and experiments and simulations confirm the practicability of our system.

We organize the rest of the paper as follows. Section II presents some related works. Section III illustrates the system architecture. Section IV describes the strategies of organizing local network and downloading data. Section V describes the RAP Incentive Scheme in detail. Section VI shows the performance of the system and the analysis for the results of the experiments. The conclusion is drawn in Section VII.

II. RELATED WORK

Cooperative download system: Cooperative download has drawn increasing attention from the research community. A system in [1] allows users in local networks to share the videos downloaded from cellular networks. However, the authors assume that all the users in the network are friends and have agreed to share their videos in advance, which is not as

practical as our work. Moreover, most of the previous works based on cooperative download assume that all the users want to share the same files [2][3]. Our system supports general data downloading instead. A user does not necessarily download the same data required by all the users. A system proposed in [4] also considers the case where not all users are interested in the same network resource. However, the authors did not take the dynamic characteristics of local network into account. Users in our system can join/leave download groups at any time. The nature of our system also signifies that the method of broadcast or multicast in WiFi networks [5][6] cannot be applied either.

Incentive scheme: In order to guarantee the cooperation among the users in a system, incentive mechanism design is of great significance. Our system has to deal with some similar problems that normal P2P systems face, including free-riding and white-washing [7]. The two main categories of incentive mechanisms are based on users’ reputation-rating schemes [8][9][10] and monetary payment schemes [11][12] respectively. However, most of the mechanisms deal with the situation in a distributed network and thus are not suitable to employ them to an unstable mobile network with a centralized management server. Outside the world of P2P, pricing mechanism is also implemented as a highly effective method to lead users to undertake some expected strategies [13]. In this paper, we propose RAP to handle the payment of clients, which is appropriate for the centralized architecture of our system.

Self-organizing network: The idea of combining different wireless interfaces to improve power performance has been investigated before. For instance, [14] made use of Chipcon and Bluetooth to reduce the power consumed when wireless terminals automatically scan and connect to an AP of WiFi. [15] reduced the number of scans for AP discovery by using a combination of Bluetooth contact-patterns and cell-tower information. However, they considered the case of connecting to WiFi access point only, and how to dynamically organize a local wireless network using multiple interfaces while keeping low power consumption is still an open question.

Data plan sharing: Only a few works have considered the problem of static data plan. [16] proposed a cost-aware caching algorithm to achieve better latency reduction and network cost reduction. [17] considered the monthly limit on data plan usage and investigate the ideal and reality of web caching on smartphones. To the best of our knowledge, INDAPSON is the first data plan sharing system combined with cooperative download on smartphones.

III. SYSTEM ARCHITECTURE

As shown in Fig. 1, our local network consists of two different layers: the Bluetooth-Management Network (BMN) and the WiFi-Transmission Network (WTN). Wireless terminals in WTN are actually also contained in BMN, however WTN is not a subnet of BMN because it is based on WiFi connections while BMN is based on Bluetooth connections.

We assume that each of the users has access to Internet via cellular data connection. If client *A* wants to speed up its

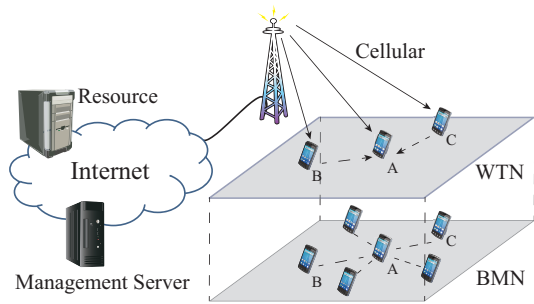


Fig. 1. One example of download group.

download process or borrow others' data traffic, it starts to construct BMN first, and asks for the help of other clients in BMN. For instance, client *B* and *C* accept the request and then *A*, *B* and *C* compose WTN. During the download process, client *B* and *C* download parts of the data from the Internet using their own cellular interfaces, and retransmit the data to client *A* through their WiFi interfaces. *A* merges the data from the other clients and thus consumes less data traffic and achieves much higher download speed. At the end of the download process, all the clients will upload their download records to a management server on the Internet, and thus the contributions of users are recorded. With the scheme of RAP applied, *B* and *C* can get virtual credits from *A* as reward. As described above, we have three kinds of clients in INDAPSON: *primary user*, *assistant user*, and *detected user*. Primary user (PU) is the client which initially requests data from internet, namely, *A* in Fig. 1. Assistant user (AU) is the client which helps primary user to download data from server, namely, *B* and *C* in Fig. 1. Detected user (DU) is the client in BMN but not in WTN. The basic assumption for the clients is that all of them can use cellular and Bluetooth/WiFi interfaces synchronously, and the function of soft AP has also been installed. Clients are connected to the Internet via cellular interface, the local network is managed by Bluetooth, and data are transmitted between clients via WiFi.

IV. SELF-ORGANIZATION AND COOPERATIVE DOWNLOAD STRATEGIES

A. Organization and Download Scheme

In this subsection we introduce the organization process of local network and the download scheme for all the mobile terminals in a download group. Strategies in detail will be shown in subsection IV-B and IV-C. A *download group* consists of all the clients in BMN and WTN. To guarantee both stability and efficiency during a transmission, the data to be downloaded is viewed to consist of segments with fixed size (except the last one). Thus we design a dynamic dissemination scheme to manage the transmission of segments. Instead of downloading a segment of data once for all, our scheme requires the terminals download several independent segments alternately. In order to reduce the occupation of buffer, all the relay clients retransmit the segments directly without any storage.

The scheme consists of six parts: the query of credits, the construction of BMN, the construction of WTN, the

initialization of download, the process of download, and the completion of download. We will describe them in detail.

The query of credits: When a client needs to download data from the Internet, it firstly gets the size of the target data. Then it connects to the management server and uploads the target data size. The server calculates the cost of the download process, and returns the cost and surplus credits to the client. If the surplus credits are enough to pay the download, the process goes on.

The construction of BMN: In this process the client initializes its BMN and becomes the PU in the network. It scans neighboring clients and checks if they have also installed the same application. If so, it sends a request message to the clients and waits for replications. We say the PU is in *scanning mode* when it keeps scanning and sending request messages. As introduced before, each of the clients always listens to the Bluetooth interface if the client is idle. We say a client is in *listening mode* if it has not joined any BMN and its Bluetooth interface is waiting for the request message. Thus the neighboring clients in listening mode can receive the requests. If the user of a client has agreed to act as an AU, the client will send a positive replication, including the information about its battery level. Then the PU will list this client as a DU in the BMN. The action of scan is operated in round, and the intervals between adjacent rounds follow heuristic scanning method, which will be introduced later. Note that the start of BMN's construction is on-demand: the process is initialized only when PU wants to download something. Moreover, this process continues and the PU stays in scanning mode during the whole download process.

The construction of WTN: After the first round of scan, the PU starts to construct WTN. The soft AP is initialized with SSID and password created randomly. Because of the limitation of mobile devices, there exists an upper bound for the number of clients that can connect to the PU at the same time. In addition, we also need to take transmission collisions into account to avoid the bottleneck effect of WiFi. Thus only part of the DUs can be invited to be AUs, and the maximum number of them is controlled according to the measuring result, which will be discussed in Subsection IV-B. If there are more than one DUs available after a round of scan, those with more dump energy will be chosen as AUs preferentially. In our scheme for the DUs selected to be AUs, the PU will invite them by sending its SSID and password via Bluetooth. According to the SSID and password received, a DU tries to connect to the soft AP operated by the PU. If the connection is set up successfully, the DU informs the PU and then be listed as an AU in the WTN.

As the members of the BMN changes with time, clients may also join or leave the WTN even if the download process has started. If an AU leaves the WTN, a DU in BMN (if it exists) will be invited to be a new AU, and then new tasks are assigned to this AU.

The initialization of download: After the construction of WTN, the PU has to initialize the download process before it can download data from the Internet. The URL of the target

data is broadcasted to all the AUs in the WTN. Then both the PU and AUs try to connect the resource server for the data. If an AU cannot connect to the source successfully, it will be deleted from WTN, and a new AU will be invited from BMN.

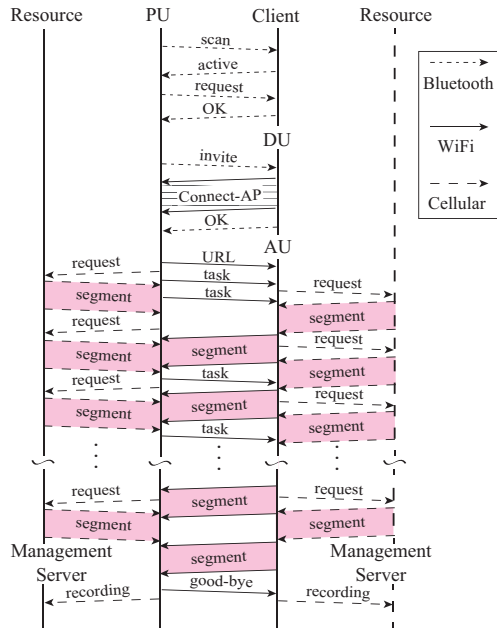


Fig. 2. The exchanged messages and segments. The process for the query of credits is omitted.

The process of download: At the beginning of this process, PU divides the data to be downloaded into segments with constant size, except that the size of the last segment is not fixed. Each segment is assigned as a *task* to a client in WTN. Every client will get task messages initially, and according to the information included in the task messages, the clients request to download corresponding data segments from the resource server. For AUs the segment will be retransmitted to PU immediately. Once PU has received a segment from an AU completely, it will assign a new task to the AU, if not all the tasks have been assigned. Thus each client will download more than one segment to complete the task. This process follows the dynamic task dissemination strategy (shown in Subsection IV-C) and repeats until all the segments of the data have been downloaded. For each segment received from the Internet or the AUs, the PU writes it to an appropriate position in the file according to its start and end position.

The completion of download: When PU has received all the segments, it will broadcast good-bye messages to all the AUs, and then send a recording message to the management server. Having received the good-bye messages, AUs will also upload their download record, including their dump energy and the number of segments they have relayed. Even if the download process is interrupted, or an AU leaves the network before it finishes downloading, the record can still be uploaded automatically after a certain interval. The management server updates its record about the contribution of users according to these recording messages, and virtual credits are exchanged according to RAP scheme. Note that since all the clients

upload their records to the management server, confliction can be detected if a malicious PU/AU uploads fake record. The whole download process is shown in Fig. 2.

B. Dynamic Self-Organization Strategy

As introduced before, to reduce the power consumption caused by monitoring the request to build local network, we use Bluetooth instead of WiFi to keep listening to the wireless channels, and a power-saving listening method is used. In order to adapt our strategy to the dynamic characteristics of mobile network, we also propose a heuristic scanning method to save more energy. Moreover, to avoid the bottleneck problem of WiFi, which will be shown in Subsection VI-C, we also need to control the number of AUs according to the real time transmission rate of cellular and WiFi. We will introduce these methods in detail.

Power-saving listening method: Even if the power consumption caused by listening to wireless channel is less than that caused by wireless transmission, it is still necessary to take power control into account. According to [15], Bluetooth costs less than WiFi even in idle case. The results of our experiments also show that on Android testbeds, in 500 minutes WiFi interface in listening mode costs 588mAh more than that costed by Bluetooth. Experiments in detail will be shown in Subsection VI-B. Thus we use Bluetooth interface to build the local management network. To further reduce power consumption, we make the clients to be intermittent sleeping: with a period of 60 second, the client listens to the wireless channel for 30 seconds, and for the rest 30 seconds it keeps sleeping. Since it is the nature of our network that fast-moving clients should not be taken as DUs, and the delay caused by this method is bounded, it is reasonable to allow the DUs to fall asleep periodically. The performance of this listening method will be measured in Section VI. It should be noticed that a client will leave listening mode once it has been a member of BMN, and thus it cannot join two different BMNs at the same time.

Heuristic scanning method: The operation of scanning Bluetooth devices costs more power than listening for request message only. Since the time taken by scanning is related to the number of neighbor clients and not fixed, we control the interval between two adjacent scans to control the power consumption in scanning mode. Since potential DUs will not change a lot in a relatively static environment (for example during a meeting time), the scanning interval can be extended if the results do not change a lot in adjacent scanning rounds. In our method, time interval will be doubled if the results of the previous two scans do not change and the interval has not reached the upper bound we have set. Otherwise it will be halved before it reaches the lower bound. Since the scanning interval is unequal to the sleeping time in the power-saving listening method, and the period of scan is not fixed, the scan and sleep of clients will not interlace with each other. Thus the maximum time needed to discover a nearby client is still bounded. With this method, our system can save more power

in relative static environment, and still be able to adapt to dynamic environment.

It should be highlighted that even if a PU has started downloading, it is still in scanning mode. Thus new clients are still requested to join BMN. Once a connection to an AU is blocked or interrupted, the PU will delete the AU from WTN, and a DU in BMN will be invited to be a new AU. Therefore in our network clients are allowed to join or leave the download group, which is mainly controlled by BMN.

Quantity Control of AUs: It is obvious that congestion in WTN will be more serious if there are more AUs transmitting data via WiFi. As what will be shown in Subsection VI-C, the average data rate of a single link in WTN will decrease with the number of AUs. Once the average data rate of WiFi is smaller than that of cellular network, transmission in local network becomes the bottleneck instead. Thus to avoid this problem it is necessary to control the number of AUs in a WTN. The main idea of this method is that, every time a new AU has joined WTN, PU measures the data rate of a single link via both cellular and WiFi. Once the rate via WiFi is no larger than that of cellular, the current number of AUs is set as the upper bound for it, which means that no new AUs will be invited unless any AU leaves the network.

C. Dynamic Task Dissemination Strategy

Since the transmission rates of clients are different from each other, for static task arrangement it is possible that the clients with higher data rates finish their tasks earlier and be idle while the others are still transmitting segments busily. In other words, to make full use of all the download links, it is necessary to apply a dynamic task dissemination strategy to handle the download process.

After PU has gotten the URL from web browser, it connects to the resource server to get the data size. Then the task to download the data is divided into small tasks, and each task requests a client to download a data segment from the resource server using HTTP. All the tasks are stored in a task queue. At the beginning of the download process, for each client in WTN, PU gets two tasks from the task queue and assigns them to the client. The reason why two tasks are initially assigned to each client is that the delay caused by waiting for the transmission of task messages can be reduced. Having received a task message, a client can get the begin and end positions of the segment, and then start to request the data from the Internet. Once a PU/AU has finished downloading/retransmitting a segment, a new task will be assigned to it. This process keeps going on until the task queue is empty. With this dynamic strategy applied, we ensure that the clients with larger maximum data rates download more segments than those with smaller rates.

For most transmission strategies in mobile network, it is necessary to take the mobility of users into account. A sudden departure of a user will cause disconnections among other users. Our dynamic task dissemination strategy also considers the retransmission of data segments in this case. For each task removed from the task queue, it will be added to a timing

table. Once a task is time-out, it will be added back to the head of the task queue, and a cancel message will be sent to the corresponding client to cancel this task. If the transmission from an AU is always blocked, then PU will delete this AU from WTN by sending a good-bye message. Thus if an AU has no access to the Internet or it leaves the local network during downloading, the transmission blocked can still be recovered with the help of the timing table.

During the implementation of our system, HTTP/1.1 methods are followed, and therefore the task dissemination strategy can be applied to any resource as long as partial content downloading is allowed by the resource server. Due to limitations of space, other implementation details such as the modification of routing table and the protection for users' security are omitted here. One can refer to our technical report [18] for discussion in detail.

V. RAP INCENTIVE MECHANISM

One of the biggest differences of our system to others is that, in many cases, only one of the users starts a co-download task while the other users are not interested in the target data. Therefore we need to provide incentives for users who help others download data, and make the exchange between static data plans and virtual credits possible. Furthermore, how to prevent malicious users from downloading data only and not helping others relay traffics is also a problem. In order to tackle these problems, we propose a *Reputation Adaptive Pricing Scheme* (RAP) based on the architecture and working process of INDAPSON.

The intuitive idea in RAP is to implement a Reputation Metric to formulate a more effective pricing scheme in a mobile energy-sensitive cooperative system with a more limited bandwidth than ordinary cooperative streaming systems. Since the records of download processes are uploaded to a management server, the server can maintain a virtual credit account for each user as well as the download and relay history of the user. If a user wants to utilize our system, he must register an account and, in the meanwhile, he will get a certain amount of virtual credits as the initial money. Based on this foundation, the RAP scheme consists of the following four stages.

Stage 1: After a PU starts a co-download task and uploads the target data size, the server firstly calculates the basic price for the download process. We set up a flat unit price p_0 for per unit data downloaded by a client, and F represents the size of the target data. Note that PU itself is also considered as a client that will get reward. Thus we can get the basic expected price P_1 at the beginning of this task:

$$P_1 = F \cdot p_0. \quad (1)$$

Stage 2: We define a simple metric to present the reputation of PU in our system:

$$M = \frac{D}{R}. \quad (2)$$

Here, D , R denotes the user's total amount of data downloaded as a PU and relayed as an AU in history, respectively. Both D

and R are stored in management server. Thus the server can calculate the adapted total price P_A to pay in this task:

$$P_A = (1 + \gamma)^M \cdot P_1. \quad (3)$$

Here, γ is the *punishment factor*, a positive constant set by the management server. Then P_A is returned to the PU.

Stage 3: If the surplus credits are enough, the PU decides whether or not to accept this price. If accepted, turn to the next stage; otherwise, the download process is stopped. The BMN is dismissed once the process is blocked at this stage.

Stage 4: After the Cooperation task is completed, all the clients will upload their download record to the management server. The server then calculate the credits that should be transferred from PU's account to AUs' accounts. For each client that has downloaded data (including the PU and the AUs that have left WTN before the download is finished), the server calculates the revised unit price for $Client_i$ ($i = 1, \dots, N$):

$$p_i = \frac{p_0}{1 + A \cdot E_i}, \quad E_i \in [0, 1], \quad (4)$$

where E_i represents the energy percentage of $Client_i$ at the beginning of the download, and A is a positive constant. The credit that $Client_i$ can get from PU is:

$$P_{C_i} = \frac{p_i f_i}{\sum_{j=1}^N p_j f_j} \cdot P_A, \quad (5)$$

where f_i denotes the actual size of data relayed by $Client_i$. Note that P_A is calculated by the server according to D and R stored locally, and thus a malicious PU cannot reduce its total expenses by uploading fake record. At last, all the download and relay behaviors are recorded by the server who updates the database of the users.

There are several advantages in RAP scheme. Firstly, we utilize a reputation mechanism to give punishment to the selfish users who are not likely to act as AU. The famous P2P system, KaZaA, also uses the same metric as (2) to rate the users. The effectiveness of this reputation mechanism is discussed in Subsection VI-F. Secondly, our system can leverage the pricing scheme to compensate AUs of less remaining energy, who will receive higher rewards per times of cooperation. Thirdly, users can deal with their dynamic data usage under static data plans with RAP. For instance, a user can transfer part of his data plan into virtual credits if he can't use up his entire data plan. Then if he would like to use more than the limited amount of his data plan in a special month, he can act as the PU and spend the virtual credits to let AUs undertake download tasks. However, in the long term, he has to keep a balance of playing the roles of PU and AU; or he will get punishment with a higher reputation adaptive price.

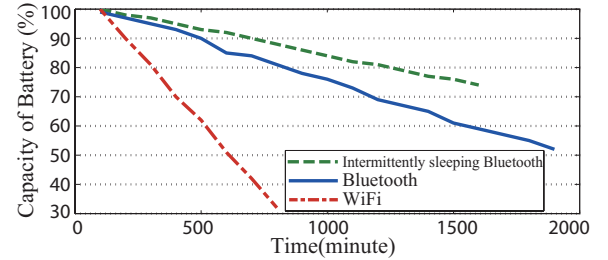
VI. MEASUREMENT AND EVALUATION

In this section, we evaluate the performance of our data plan sharing system by conducting experiments on Android testbeds consisting of Android smartphones. During the tests, we used 1 to 7 Android smartphones, and all of them are Nexus 4 with Android Jelly Bean (4.2.2) as their operating

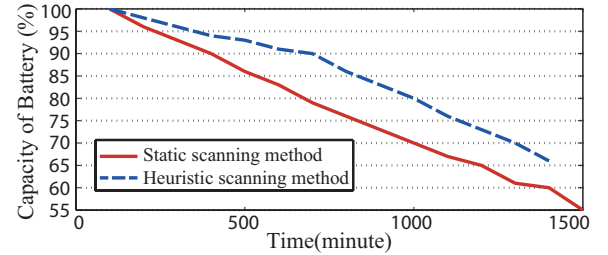
systems. Each of these smartphones is equipped with a 1.5GHz CPU and 2G RAM. The type of cellular network is 3G, which has been widely used around the world.

A. Measurement of Power Consumption in Listening Mode

To test the benefit of the power-saving listening method, we tested both the cases with or without intermittent sleeping. For the previous case, the period was 60 seconds, during which the client listens to the wireless channels only for 30 seconds. Moreover, we also tested the case where WiFi interface was used to listen to the request message.



(a) Power consumption in listening mode.



(b) Power consumption in scanning mode.

Fig. 3. Power consumption test. The capacity of the battery used is 2100mAh, and the remaining capacity is recorded every 10 minutes.

Fig. 3(a) shows that WiFi interface cost more energy when we kept it waiting to accept request messages. For the case of power-saving listening method, its power consumption was less than that of normal Bluetooth. It can also be seen that even if sleeping time made up half of the total test time, the energy consumed by intermittent listening was still larger than half of that needed by normal case. The reason is that initializing the listening process of Bluetooth also cost a small amount of energy. The result indicates that, with the method of intermittent sleeping, the power consumption is further reduced, which confirms the benefit of the power-saving listening method. The only price is that the delay to construct a local network is increased. However, this delay is bounded and acceptable.

B. Measurement of Power Consumption in Scanning Mode

We evaluated the power consumed during the scanning process of Bluetooth, and the effectiveness of the heuristic scanning method was also tested. Since a client is in scanning mode during download processes only, we forced a smartphone to keep scanning to get a evident result. Both the cases with and without heuristic scanning method are tested. The original interval of scan was set to be 10 seconds during the test, and the lower bound of it was also 10 seconds while the

upper bound was 40 seconds. We wrote a program to simulate that no client left or joined the local network during the first 750 minutes, and after that surrounding mobile terminals kept changing during each scan. Thus with heuristic scanning method applied, the average interval of scan was 40 seconds before the 750th minute, and became 10 seconds after that. If the method was not applied, the interval was 10 seconds constantly. The capacity of the battery used is 2100mAh.

As shown in Fig. 3(b), during the first 750 minutes, the power consumption was reduced by heuristic scanning method. For the rest time, since the interval of scanning was reduced back to 10 second, the performance of device discovery was improved while the power consumption increased. In another word, more energy was saved during the time when almost no other users moved around, and system performance could still be recovered when surrounding terminals changed frequently. Since a longer interval between two scans does not affect the performance of device discovery a lot in an almost static environment, it is reasonable to save more energy while keeping original property.

C. Measurement of WiFi Network Congestion

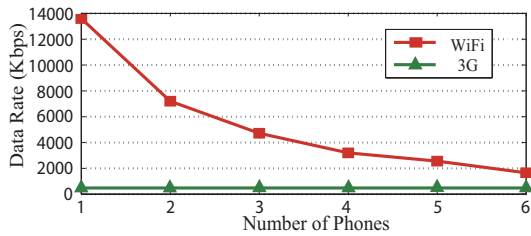


Fig. 4. Data rate of WiFi with collisions. Each value shown in the figure is averaged over five experiments.

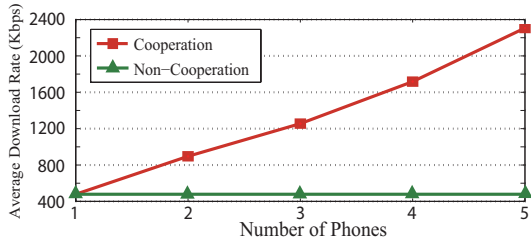


Fig. 5. Average download rate of PU with different number of phones. Each value shown in the figure is averaged over five experiments.

Before the evaluation of INDAPSON, it is necessary to measure the maximum capacity of WiFi. Theoretically, the data rate of the WiFi link between AUs and PU should not be smaller than that of 3G to ensure that an AU can timely forward the data it downloads through 3G. If not, the effectiveness of our download system would be influenced by the WiFi network congestion. Here we test the transmission rate of WiFi when the number of AUs varies. During the test one to six smartphones were connected to a PU and transmitted data to it simultaneously.

Fig. 4 shows that the WiFi data transmission rate decreased as the number of AUs in the download group increased. Then it was possible for WiFi data rate to be smaller than that of cellular if there existed more AUs or more severe background interference. Therefore it is obvious that quantity control of

AUs is necessary in our system. It can also be seen that even when there were 5 other WiFi links which introduce strong collisions, the average WiFi data transmission rate still reached about 1650Kbps, which was much higher than that of 3G. Thus even with the strong WiFi interference such as 5 other busy WiFi links, the data rate of WiFi is still high enough to retransmit the data received from the Internet via 3G, which confirms the scalability of our system in a not too large download group.

D. Evaluation of INDAPSON Download Rate

In this subsection we evaluate the download performance of our system. We varied the number of AU from 1 to 4 to test the average download rate. During the experiment all the smartphones had access to both 3G and WiFi connections. The rate of 3G fluctuated between 420Kbps and 510Kbps, and WiFi supported up to 13600Kbps. The size of the data downloaded was 299MB.

Fig. 5 shows the average download rate of PU with different number of smartphones in the download group. It can be seen that the rate increases with the number of phones linearly. Since the data rate of WiFi is always larger than that of 3G in this case, the download data rate for PU is proportional to the sum of the 3G download rates of all the phones. Although the momentary rate of 3G changes with time, the average 3G rate does not vary a lot. Therefore for each new AU in the download group, the average download rate increases by about 457Kbps. By comparing the average download rate with and without cooperation, we can see that taking the advantage of the idle 3G data interfaces of the AUs improves the download rate of the PU greatly. The essence of the cooperation download system is that more idle 3G data connections of the base station is used during the download process, and the total data size of the segments downloaded do not change. Thus INDAPSON can linearly increase the download rate with constant average 3G data rate.

E. Evaluation of Dynamic Task Dissemination Strategy

In this subsection we test the effectiveness of dynamic task dissemination strategy. During the experiment we used one AU to help a PU to download a file with size 8332KB. The cellular interface of the AU is 2.5G (EDGE), while the PU still uses 3G to download segments.

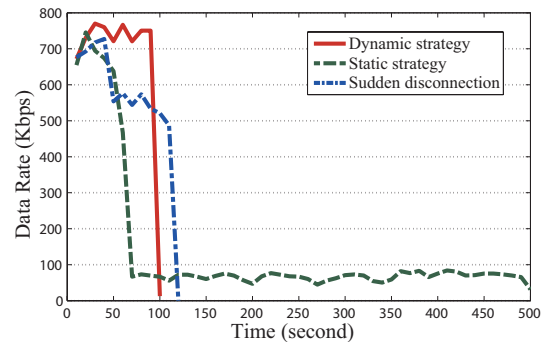


Fig. 6. Comparison between dynamic and static task dissemination strategy.

At the first part of our experiment, we applied a static task dissemination strategy, which means that the amount of

segments downloaded by PU and AU were equal. As shown in Fig. 6, after about 70 seconds the PU finished downloading its own part while the AU was still downloading segments busily. The total time consumed to download the file was 500 seconds. At the second part of the experiment, we applied the dynamic task dissemination strategy to manage the download of segments. The result shows that the download rate did not decrease around the 70th second, and downstream rate was kept to be almost stable until the download was completed. The total time used to download the file was no larger than 100 seconds for this case, which is much shorter than that needed by static strategy. At the last part, to simulate a sudden disconnection in the case of dynamic strategy, we cut off the download link of the AU at the 50th second. We can see that the file was still downloaded completely in 130 seconds.

Comparing these results, we can draw a conclusion that the usage of dynamic strategy can make full use of every transmission link, and the download processes of all the clients can finish at almost the same time. Moreover, dynamic strategy is also able to detect the sudden departure of AU, and thus the possible interrupts caused by sudden disconnection are avoided. These results show that the dynamic task dissemination strategy ensures the stability of our system, and the average download rate is also increased.

F. Evaluation of RAP Incentive Scheme

1) *Setup of Simulation:* In our simulation, we divide time line into rounds and, in each round, users have 3 states: shutting down our application, running our application without an attempt to download and running our application with an attempt to download. We denote the states as S_0 , S_1 and S_2 , respectively. In each round, if a user is in S_2 , he will receive a list of K users who are nearest to him currently. K is normally distributed with a mean of 20 and a variance of 5 in our experiment. However, the K users are not necessarily running our application on their mobile terminals. Among the K users, only N users are running the system on their phones and can be selected as the DU/AUs. In order to simulate users' behaviors in our system as accurately as possible, we consider three aspects: 1) users' nature; 2) an economic model considering revenue and cost in a round; 3) a response mechanism. To simplify our analysis, we consider PU and AU only during the simulation because the ratio between DU and AU changes little on average and thus does not affect the result of the simulation.

Firstly, in order to simulate the different users' nature, we set two types of users, denoted as *Ordinary Users* and *Malicious Users*. Both of the two types of users may act as PU or AU in our system. Malicious Users also undertake free-riding and white-washing while Ordinary Users do not. We implement a random algorithm to set up the basic behavior parameters of the users. We denote the basic probability of a user in S_1 and S_2 as P_{relay} and $P_{download}$, respectively. The average P_{relay} of Malicious Users is much smaller than that of Ordinary Users. To simulate the dynamic data usage, the two parameters change randomly as the time goes by.

Secondly, we implement an Acceptance Probability [19][20] as another factor to depict the users' behaviors. The probability to undertake a behavior is $A(G, C) = 1 - e^{-\alpha \cdot G/C}$. G, C denote the revenue (gain) and the cost of this service respectively. α is a positive constant. As to PU, G_{PU} is defined as the product of cooperative bandwidth and the target data size; C_{PU} is P_A in (3) plus the charge of data usage by the ISP. As to AU, G_{AU} is defined as P_{C_i} in (5); C_{AU} is the predicted energy consumption plus the charge of data usage by the ISP. This model is similar to the Cobb-Douglas Demand curves [19], widely used in economics.

Thirdly, the two types of users will respond differently if they fail to set up a cooperative download task because of the too expensive price, which is more than the current amount of their account credits or much more than they expected. In terms of Malicious Users, the way for them to constantly benefit from our system with a small level of contribution is to constantly register as new users. In terms of Ordinary Users, they will not register again. They adjust their frequencies in S_1 by improving their probabilities of S_1 with a certain amount, denoted as P_Δ . But P_Δ will decrease to 0 linearly as the time passes by. Besides, the actual behaviors of the users are also influenced by battery energy level E .

Therefore, in a round, the actual probability of two types of the users in S_2 equals to: $f_{S_2} = P_{download} \cdot A(G, C)$. And the actual probability in S_1 equals to: $f_{S_1} = (1 - f_{S_2}(E)) \cdot (1 - e^{-\beta E}) \cdot (P_{relay} + P_\Delta) \cdot A(G, C)$. As to Malicious Users, P_Δ equals to 0.

We establish 100 users and set the users' cellular bandwidth as a normally distributed value with a mean of 500Kbps and a variance of 50Kbps. The data size is set to be a normally distributed value with a mean of 20MB and a variance of 6MB. The energy consumption E starts as 1 and decreases as the time goes by and users randomly recharge. A in (4) is set to be 1. The initial money in a new user's account is 100 and the flat price for per unit data is 1. Furthermore, we assume that users have different data plans (the mean is 2GB in our simulation). In a month, if they use more data than the limited amount of their data plans, the exceeding part will be charged much higher than the average price per unit of data in the tiered data plan.

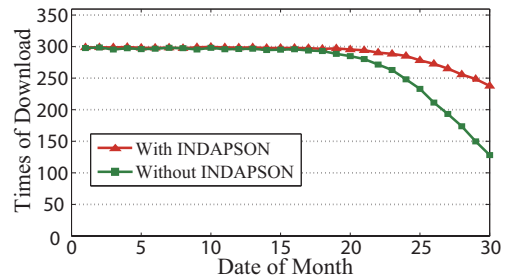
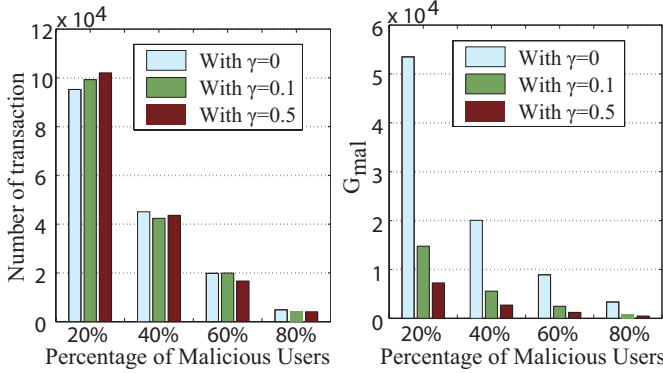


Fig. 7. The times of download per day in a month.

2) *Results of Simulation:* In Fig. 7, the number of actions of download per day decreases at the end of a month. The reason is that, since some users may use up all the data in their data plans and have to pay high price to download extra

data, their motivation for downloading is suppressed. With the help of our system, the problem will be mitigated because they can leverage others' data plan and thus avoid being charged with a much higher price. However, we still fail to keep the times of download equal to that of the beginning of a month because there are fewer users willing to act as AUs in the end of a month.



(a) The influence on Ordinary Users. (b) The influence on Malicious Users.
Fig. 8. The influence of the punishment factor γ .

We set the percentage of Malicious Users as 20%, 40%, 60% and 80% respectively and set punishment factor γ as 0, 0.1, and 0.5 to see the effect of RAP. We use the number of transaction among Ordinary Users as the metric to see how RAP affects them. From Fig. 8(a), we can see that the different punishment factors have trivial influence on the number of transactions while the percentage of Malicious Users seriously impacts the number. Furthermore, in terms of the Malicious Users, we define a metric G_{mal} to measure the benefits of a Malicious User: $G_{mal} = \frac{\sum \bar{B}_i \cdot F_i}{N_{wash}}$, where \bar{B}_i denotes the average bandwidth and F_i represents the target data size in i th cooperation task where a Malicious User acts as the PU. N_{wash} is the total times of white-washing performed by this Malicious User. G_{mal} represents the average benefits that a Malicious User gets every time he registers as a new user. From Fig. 8(b), we can see clearly that G_{mal} with RAP is much less than that without RAP. Furthermore, G_{mal} decreases as γ increases. That's because the actual price paid by the PU is an exponential function and the punishment will be much more rigorous with a larger γ .

It can also be shown that in our system a user with higher bandwidth has more credits as the reward paid by the PUs, and the AUs of less energy remaining are compensated properly. Due to limitations of space, we omit this part of simulation. One can refer to our technical report [18] for detail.

VII. CONCLUSION

In this paper we proposed a data plan sharing system based on self-organizing networks, which allows users to better manage their data traffic in a fixed data plan. In our system, with power-saving strategies applied, multiple wireless interfaces are used to construct a two-layer local network, which adapts to the dynamic characteristics of mobile terminals. Part of the users in such a network can act as relay nodes to help the users

who initiate download processes. Thus data are downloaded via multiple download links from the Internet, and the total download rate is improved. Moreover, with RAP applied in our system, users are able to exchange their data traffic for credits by helping other users download data, which confirms the operation of our system. Numerical experiments show that our system can accelerate download speed distinctively, and the results of simulation also show that our system can help users manage their data traffic effectively.

VIII. ACKNOWLEDGMENT

This work is supported by NSF China(No.61325012, 61271219, 61221001); Shanghai Basic Research Key Project (No.13510711300, 12JC1405200, 11JC1405100); China Ministry of Education New Century Excellent Talent (No.NCET-10-0580); National Key Project of China (No.2012ZX03001009); Doctoral Fund of Ministry of Education of China (No.13005160)SEU SKL project (No.2012D13).

REFERENCES

- [1] L. Keller, A. Le, B. Cici, H. Seferoglu and C. Fragouli, "MicroCast: cooperative video streaming on smartphones," in *Proc. of MobiSys'12*, pages 57-70, 2012.
- [2] M. Ramadan, L. El Zein and Z. Dawy, "Implementation and evaluation of cooperative video streaming for mobile devices," in *Proc. of IEEE PIMRC'08*, pages 1-5, 15-18, Sept. 2008.
- [3] H. Seferoglu, L. Keller, B. Cici, L. Anh and A. Markopoulou, "Cooperative video streaming on smartphones," in *Proc. of Allerton CCC'11*, pages 220-227, 28-30, Sept. 2011.
- [4] G. Ananthanarayanan, V.N. Padmanabhan, L. Ravindranath and C.A. Thekkath, "COMBINE: leveraging the power of wireless peers through collaborative downloading," in *Proc. of MobiSys'07*, pages 286-298, 2007.
- [5] L. Man-Fung and S.H.G. Chan, "Broadcast-Based Peer-to-Peer Collaborative Video Streaming Among Mobiles," in *Proc. of Broadcasting, IEEE Transactions*, vol.53, no.1, pages 350-361, Mar. 2007.
- [6] C. Chi-Yin, L. Hong Va and T.S.C. Alvin, "Distributed group-based cooperative caching in a mobile broadcast environment," in *Proc. of IEEE MDM'05*, pages 97-106.
- [7] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," in *Proc. of IEEE JSAC'06* 24(5):1010C1019, May 2006.
- [8] B. Cohen, "Incentives build robustness in bittorrent," in *Workshop on P2P Econ*, Vol. 6, pp. 68-72, 2003.
- [9] K. Walsh and E.G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in *Proc. of NSDI'06*, pages 1-1, Berkeley, CA, USA, 2006.
- [10] S. Song, K. Hwang and R. Zhou, "Trusted P2P Transactions with Fuzzy Reputation Aggregation," in *Proc. of Internet Computing, IEEE*, vol.9(6), pages 24-34, Nov. 2005.
- [11] V. Vishnumurthy, S. Chandrakumar and E.G. Sirer, "Karma: A secure economic framework for p2p resource sharing," in *Workshop on P2P Econ*, Berkeley, California, 2003.
- [12] R. Jurca and B. Faltings, "Reputation-based pricing of p2p services," in *Proc. of the ACM SIGCOMM workshop on P2P Econ*, pages 144-149, 2005.
- [13] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang, "TUBE: Time-Dependent Pricing for Mobile Data," in *Proc. of SIGCOMM'12*, August 13C17, 2012, Helsinki, Finland.
- [14] T. Pering, V. Raghunathan and R. Want, "Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup," in *Proc. of VLSID'05*, pages 774-779, 2005.
- [15] G. Ananthanarayanan and I. Stoica, "Blue-Fi: Enhancing Wi-Fi Performance using Bluetooth Signals," in *Proc. of MobiSys'09*, pages 249-262, 2009.
- [16] P. Cao and S. Irani, "Cost-Aware WWW Proxy Caching Algorithms," in *Proc. of USITS*, Monterey, California, Dec. 1997.
- [17] F. Qian, K.S. Quah, J. Huang, J. Erman, A. Gerber, Z. Mao, S. Sen and O. Spatscheck, "Web caching on smartphones: ideal vs. reality," in *Proc. of MobiSys'12*, pages 127-140, 2012.
- [18] T. Yu, Z. Zhou, D. Zhang, X. Wang, Y. Liu and S. Lu, "INDAPSON: An Incentive Data Plan Sharing System Based on Self-Organizing Network," <http://iwct.sjtu.edu.cn/Personal/xwang8/paper/indapson.pdf>.
- [19] H.R. Varian, "Intermediate Microeconomics: A Modern Approach," Norton, New York, March 1999.
- [20] O. Ileri, D. Samardzija, and N.B. Mandayam, "Demand responsive pricing and competitive spectrum allocation via spectrum server," in *Proc. of IEEE DySPAN'05*, pages 194-202, Nov. 2005.