2014 IEEE 28th International Conference on Advanced Information Networking and Applications

CollabAssure: A Collaborative Market Based Data Service Assurance Framework For Mobile Devices

Bhanu Kaushik*, Honggang Zhang†, Xinyu Yang‡, Xinwen Fu* and Benyuan Liu*
*Department of Computer Science, University of Massachusetts, Lowell MA, USA 01854.
Email:{bkaushik, xinwenfu, bliu}@cs.uml.edu
†Department of Computer and Information Science, Fordham University, Bronx NY, USA 10458.
Email: honggang@storm.cis.fordham.edu
‡Department of Computer Science and Technology, Xi'an Jiaotong University, P.R. China
Email:yxyphd@mail.xjtu.edu.cn

***Abstract*—Concomitant to the growing popularity of Internet enabled mobile devices such as smartphones, tablets, PDAs, portable media players etc., however, are the concerns about availability of Internet access points for these devices. Mobile users often either overpay for service availability such as (3G or LTE) or suffer incapability of accessing Internet services due to limited hardware resources (3G or LTE) or exhaustion of carrier enforced data plans. In this paper we introduce *CollabAssure*, an auction based, ad-hoc market model assuring service for users with no Internet access capability. *CollabAssure* framework provides service assurance through opportunistic ad-hoc networks formed by spatio-temporally co-existing mobile users. The system allows users to *"sublet"* their surplus data plans to the users without Internet access. We discuss the design and implementation of *CollabAssure* technology in *Android* framework. Our simulation results advocate the success of this approach on real world traces, where mobile users need to participate in auctions for achieving on-demand and low-cost data service.**

***Keywords*-Service Assurance; Collaborative data sharing; Wireless networks; Auctions.**

I. Introduction

Availability of mobile Internet forms the backbone of current mobile data communication networks. The emergence of new mobile and wireless networks offers opportunities to expand traditional mobile Internet-based applications. The rise of such opportunities has brought a concomitant risk of service hindrance to users. Thus, satisfying the growing user needs is an encumbrance to cellular service providers.

To cater the growing needs of Internet user base, cellular network providers offer a wide coverage, enabling ubiquitous wireless data access services. As service comes with a price, the cellular service providers only allow users to subscribe to either fixed data limit under which they experience exhaustion of limited data before the end of billing cycle or underutilization of the overall available limits at end of billing cycle. Alternatively, users choose "pay-per-use" schemes, where they end up paying huge charges for the services thus used. Among other major challenges faced by mobile Internet users are unavailability of hardware (3G or LTE), unavailability of access points, service outages, network and server overloads. Thus, from these scenarios, users suffer from either overpayment for higher service availability or are rendered incapable of accessing web services. Due to all these issues and constraints, at a given instance of time, the overall mobile data user-base is virtually divided two sets, one with Internet access capabilities and other with no Internet access capability.

In our previous work [1], we introduced the design and implementation of a collaborative data sharing system, *"SmartParcel"*, to address the issue of cellular data offloading and service assurance, by harnessing spatio-temporal network of coexisting mobile devices namely, *"The Familiar Strangers"* [2]. Such decentralized mobile ad hoc networks are characterized by their completely autonomous, dynamic, self-organized and ubiquitous nature. By providing a mobile operating system approach, *SmartParcel* could be easily integrated into current mobile platforms. Though promising, SmartParcel might be vulnerable to the *"Tragedy of Commons"* problem [3], i.e., in a typical scenario all the users are waiting for a generous user to share data. Major reason for the unwillingness of a user to participate is lack of any incentive for the user to provide data to other nodes in the proximity.

In this paper we propose, *CollabAssure*, an auction based, ad-hoc market assuring service for users with no Internet access capability. *CollabAssure* framework provides service assurance through opportunistic ad-hoc networks formed by spatio-temporally co-existing mobile users. The system allows users to *"sublet"* their surplus data plans to the users without Internet access. In event of any user requesting service in an opportunistic network, *CollabAssure* allows users to participate in an auction where the nodes offering service submit their bids to nodes requesting services, resulting in formation of an ad-hoc market where both buyers and sellers request and offer services respectively. We design and implement the *CollabAssure* technology in *Android* framework and perform trace based simulations to evaluate its performance on real world traces.

1550-445X/14 $31.00 © 2014 IEEE
DOI 10.1109/AINA.2014.25

Our trace based simulation results advocate a higher Internet data service is ensured by using the ad-hoc market approach for data sharing. We observe that data refresh rates, buyer participation probability and seller participation probabilities have a significant impact on the resulting market utility. Even with a limited participation of users in both, buyer and seller roles, the market utility is significantly improved. As user activity is very high during the day time more number of trades are observed, whereas number of trades during early morning and late night hours are relatively low. Our results also indicate that user's social activity plays a vital role in the utility maximization.

The rest of the paper is organized as follows. Section 2 presents the motivation and related work for *CollabAssure*. In Section 3, we present the straw man design followed by the ad-hoc market approach in Section 4. Section 5 highlights the system design and implementation in Android. Simulation set up and results are presented in Section 6 and finally we conclude in Section 7.

II. Related Work

A. Data Offloading and Sharing

Increasing number of devices and high reliance on cellular data bring the spotlight on cellular data offloading schemes. Addressing this by providing rapid deployment and trouble-free operation using large scale Wi-Fi networks solutions, Alvarion [4] and Cisco [5] ensure a superior user experience and satisfaction with high quality Wi-Fi service by enabling carriers to optimize 3G and LTE network service and moving Wi-Fi network from an untrusted network to a trusted and integral part of a carrier's network. However, Han *et al.* [6] address the issue of information delivery by target-set selection in the social context and Lee *et al.* [7] allow data dissemination through Wi-Fi in case of data bottlenecking or service delays. Incumbent to these approaches are major modifications in state of the art hardware and software technologies entailing huge infrastructure changes and high adoption costs. Adopting these technologies do not answer to the question of heterogeneity of application data transferred over the cellular networks. Also, these techniques do not provide any incentive to the user, application developers, etc. On the contrary, *CollabAssure* being an *"One-for-all"* solution, is a multi-incentive approach, assuring high mobile application data availability, better service for users incapable of accessing Internet and above all, offers a monetary sublimation of users overpaying for carrier enforced data plans.

B. Auctions

In economic theory, an auction may also refer to any mechanism or set of trading rules for commodity exchange between participants. The modus operandi of the auction forms the basis of classification of the auction mechanism. In any auction, the following properties are sought: *a*) *Auction Correctness*: Assuming all bidders act honestly, the correct winner is identified by the auction process. *b*) *Bid Confidentiality*: The bid amounts are not revealed to any bidders. *c*) *Auction Fairness*: Once submitted the bids cannot be changed or repudiated after submission. Safekeeping these requirements, to internalize the externalities of the pricing of direct and external costs of resource usage in grid, cloud, shared network resources and services has been the focus of researchers over couple of decades [8]. In the literature, certain examples can be found where auctions have been used to allocate computation and communication resources in grid or G-Commerce [9]. Auctions are also widely used in network resource sharing, specifically, in congestible networks [10], where resources are identified as ones which can be used by more than one person but increasing usage degrades their quality or only a limited users have access to the resources [11]. In such networks, it is compulsory to offer incentives to selfish nodes to forward the traffic of their peers for a better performance. *CollabAssure*, targeting pricing of network resources, focuses on achieving two different goals, namely, reaching a maximum revenue for the network and manging the allocation of resources efficiently. Hence, targeting the maximization of overall network revenue and utilization of network resources serves as the main goal of this approach.

III. CollabAssure Straw Man

The Straw Man of *CollabAssure* is illustrated in Fig. 1. The system consists of mobile devices such as smartphones and tablets. The current policies of the cellular services providers enforce data plan requirements for smartphones in the U.S., whereas, they are not mandatory for the tablet users. Such tablet users are often rendered incapable of using the Internet services when Wi-Fi hotspots are unavailable. Also the smartphone users suffer the unavailability of Internet services during service outages or sometimes because of the exhaustion of the carrier enforced monthly data limits. Hence, these polices force certain set of users to increase their monthly data limits at an extra cost. These users are not able to exhaust the complete allocated share of the data services which are typically renewed by the end of the billing cycle. Such users eventually end up paying for the services which they did not use or partially used. *CollabAssure*, in such typical settings allow these users to "sublet" their data plans to the users in need of data service. Such a revenue model serves as an *"One-for-All"* solution for all the users where, the users with abundant data services are compensated for the over payment and the ones in need of data services are ensured services. *CollabAssure* system inherently increases the services offered by various Internet based mobile applications of which the user is deprived of in event of the unavailability of access points. All these factors advocate the benefits of adopting the *CollabAssure* by the mobile application developers, OS vendors and the users.

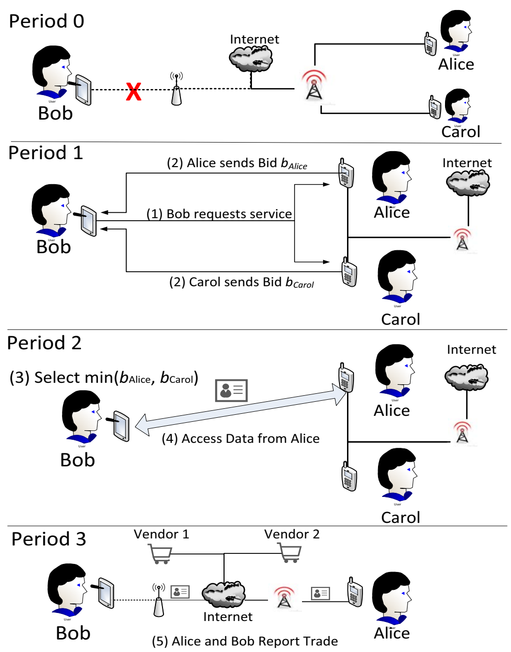


Figure 1: CollabAssure: Straw man and typical use case scenario

In a typical use case of *CollabAssure* at period 0, *Alice* and *Carol* enable the CollabAssure mode on their devices and set a data threshold for their own use. We assume that both *Alice* and *Carol* are subscribed to a 2GB monthly data plan of which 1GB is reserved for use by themselves and remaining 1GB could be sublet to other users. *Alice* and *Carol* also have full freedom to modify these limits during the complete monthly cycle. Also suppose, another user *Bob* owns a Wi-Fi enabled tablet device and *Bob* has enabled *CollabAssure* service on his device. Again suppose that, in a typical subway, both *Alice* and *Carol* enjoy the data services whereas, *Bob* cannot, because of the unavailability of Wi-Fi access points. The setup is highlighted in Fig. 1. In such a setting, *Bob* can access Internet on agreement to pay either *Alice* or *Carol* for the procurement of the Internet services. At period 1, *Bob* requests for service from nodes its in proximity. When CollabAssure services on *Alice's* and *Carol's* device receive this request, each generate a bid and communicate it to *Bob's* device. Now *Bob* picks the user with lowest bid and requests data at period 2. Once the contract is set, both *Bob* and *Alice* have Internet access and hence collateral benefits to both the users are ensured. As *Carol's* bid was higher, she does not participate in the transaction and will not earn any utility from this trade. At period 3, after the completion of the service both, seller (*Alice*) and buyer (*Bob*), report their trade agreement to there respective vendors [12] and [13]. On monthly basis, *Bob* is charged for the procured services and *Alice* is compensated for providing the data services.

IV. Ad-hoc Market Approach

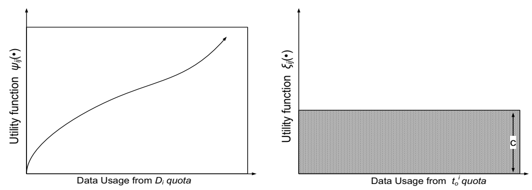
This section describes the auction based ad-hoc market approach, where all users continuously participate to buy or sell mobile data services. We use an auction strategy similar to one described in [14]. At any instance of time, for a given set of N users, arranged in an "familiar stranger" network setup, we assume that every user strives to maximize its own utility and can communicate to every other user in a full-mesh topology. For the rest of the paper we will be using the following terminology:

- *Total Data Count* (D_i): Total number of data units available for auction by seller *i*.
- *Bid* (b_i): The valuation expressed by bidding seller *i*, for each unit of data.
- *Satisfaction Count* (x_{ij}): Amount of data units buyer *i* consumes for downloading content from a seller *j*.
- *Grant Count* (y_{ji}): Amount of data units granted from seller *j* to buyer *i*, i.e., amount that user *i* uses to satisfy user *j*'s needs.
- *Usage Vector* (x_i) = $\{x_{ij}\}$, $j \in [1, N]$
- *Served Vector* (y_i) = $\{y_{ij}\}$, $j \in [1, N]$

From the definitions of *Satisfaction Count* and *Grant Count*, buyer *i* can satisfy its needs through seller *j* only if seller *j* grants the corresponding amount of data units, i.e., y_{ji}= x_{ij}. Now we define the Usage and Served matrices as $\mathbf{X}$ = $\{x_i$: $i \in [1, N]\}$ and $\mathbf{Y}$ = $\{y_i$: $i \in [1, N]\}$ respectively.

In the current scenario we assume that a mobile data user S_i, is offering data service at a marginal cost b_i per unit of data. Suppose there exists another user B_j with no available Internet connection (or one paying a much higher service cost) is requesting service. We allow user S_i to "sell/sublet" its data plan to any other user with a limited data connectivity and he is unaware of the utility of the user B_j. We also assume that a single demand from user B_j, is drawn from a continuous distribution with density $f(\theta)$ and has a finite mean. Here we reiterate that *CollabAssure* targets maximum utilization of user's data service plan. In the event of a lost connection (eg. Bluetooth out of range) between buyer and seller, unmet demand is lost, resulting in the margin being lost (to the seller), but with no additional penalty to the buyer. In this case, the *CollabAssure* framework deems it as "no-trade" and the buyer will not be charged for the trade. This loss is incurred by the seller. In general commodity market, this might not be a fair approach, but in the cellular data transaction model, such as *CollabAssure*, where the transactions are usually very small, it can be deemed as a cost of operation.

Each user *i*, sets a threshold on data limit for personal use, we call the total monthly data limit T_i and t_o^i as the threshold for user's own data needs. Hence, total data available for auction at user *i*, D_i is evaluated as D_i = T_i - t_o^i. Assuming t_i as data used by user *i* at any point of time, Fig. 2 shows the



(a) Usage greater than threshold (b) Usage less than threshold

Figure 2: Variation of satisfaction function with usage

behavior of the utility function with usage over the month, whereas Fig. 2(a) shows the scenario when usage $t_i > t_o^i$ and when $t_i \leq t_o^i$ in Fig. 2(b) where both t_i and $t_o^i \leq T_i$. Hence, for any user user *i*, the total amount of data used in auctions cannot exceed the monthly threshold, i.e.,

$$\sum_{j=1}^{N} y_{ij} \leq D_i \tag{1}$$

Also, the total amount of data used by user *i* for himself and used in auction cannot exceed the monthly data limit. This is represented as follows:

$$\sum_{j=1}^{N} y_{ij} + t_o^i \leq T_i \tag{2}$$

To design an effective bidding strategy it is intuitive that, when a user has very less data left and he is participating in auction during day time, a higher bid is expected. On the contrary, in an early morning trade when data available is relatively higher, a lower bid is expected. To model such a behavior we set the bid b_i for each transaction to be a function of the ratio of available data (δ_i) and current time of the day[1] represented by the function (λ) as follows:

$$b_i = \beta(\lambda, \delta_i) \tag{3}$$

For *CollabAssure*, we choose a simple function, $\beta(\lambda, \delta_i) = \sqrt{\lambda \times \delta_i}$, where $\lambda, \delta_i \in [0, 1]$. One may choose a more complex bid generation function, but in this case we keep it very simple. As any transaction would cost user's battery power, which is much needed during the day, we pick higher values of λ during different hours of the day and lower for the early morning and late night hours. Also, the amount of data left with the user decreases with successful trades, so we use the δ_i (ratio of data available), as a factor in bid generation. For any seller *i*, δ_i is computed as δ_i = D_i / T_i.

A trade $\gamma_k(x_{ij}, b_i)$, occurs when user *i* consumes x_{ij} data units for downloading content from user *j* at b_i unit price. Also we define $\Gamma = \{\gamma_k\text{: } (\forall\ k \mid k \in [1, \mathbb{N}])\}$ as the set of all trades, where $\mathbb{N}$ is the set of all natural numbers. Considering a general case where user *i* obtains different utility with each trade with user *j*. For any user *i* and *j* in market, the utility can be characterized as:

- A constant utility is earned by a user from utilizing resources for itself. Let $\xi_{ij}(x_{ij})$ be the utility earned by user *i* when he uses data for himself[2].

$$\xi_{ij}(x_{ij}) = \begin{cases} 0 & \text{when } i \neq j \\ c & \text{when } i = j \end{cases} \tag{4}$$

- The utility earned by using resources from any nearby user. Let $\Upsilon_{ij}(x_{ij})$ be the utility of user *i* by satisfying its data needs from user *j*. As a user is charging for the data units he shall not be using, hence the function $\Upsilon_i(\cdot)$ is a monotonically increasing utility function associated with each trade.
- The utility earned by subletting resources to another user. Let $\Psi_{ij}(y_{ij})$ be the utility of user incurred by providing resources to a user *j*.

Hence, when the conditions in (1) and (2) are met, the characterized utility function for any user *i* can thus be written as:

$$\Pi_i(x_i, y_j) = \sum_{j=1}^{N} (\Upsilon_{ij}(x_{ij}) + \Psi_{ij}(y_{ij}) + \xi_{ij}(x_{ij})) \tag{5}$$

We use a very simple model for characterizing this ad-hoc market, where the perceived utility of S_i can thus be characterized by the function $\Pi_i(\cdot)$. The proposed model brings a higher utility and revenue for any successful trade, than no trade.

V. System Design and Architecture

A. Architecture

Delay-Tolerant Networks (DTNs) [15] target the interoperability between and among challenged networks for delivering data units from a sender to a receiver in the presence of opportunistic connectivity using different transport protocols. *CollabAssure* harnesses the concepts of node discovery and opportunistic data delivery from the DTN architecture, to allow mobile devices to participate in ad-hoc markets for collaborative data sharing. We propose the CollabAssure design for android but its modular design could easily be integrated into other available mobile operating systems. Various components of *CollabAssure* framework are shown in Fig. 3. The framework has the following components:

1) ***Network Interface Manager:*** We classify different network technologies as classified as *a*) *WAN-facing or Hi-Cost interfaces (3G, LTE, GPRS, etc.)*, and *b*) *Ad Hoc facing or Low Cost (WiFi and BlueTooth)*. As android devices offer multiple network technologies, so

[1]As any trade incurrs battery cost, which is much needed during the day time as compared to late night and early morning hours, we use current time of the day to model this factor.

[2]Behavior of ξ_{ij} is independent of *j*. We use this notation to keep it consistent with other utility functions

to manage all connections we use network interface manager. Controlled by Central Control Manager, it is implemented as an internal service responsible for managing network connections on all interfaces.

2) ***Auction Manager***: Determines the perfect bidding price for per unit data at a given instance of time, when requested by the central control manager. A large number of factors govern the cost for the data download operation, namely, current battery level, available data limit, days till next billing cycle, hour of the day etc. For this we use current data limit to determine current bid.

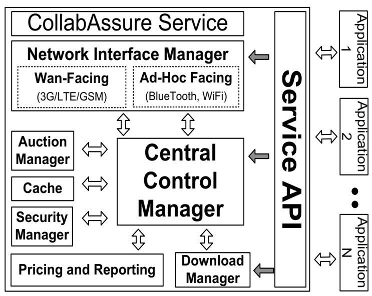


Figure 3: System Architecture of CollabAssure Framework.

3) ***Cache***: The central control manager maintains a data cache where it keeps data associated with each data download. This cache is maintained as an Sqlite data base which is available in the android framework. We use the following schema <*AppID , TimeStamp, Data*>. This cache helps user to maximize its utility by using cached data to multiple devices (buyers) at the same time.

4) ***Central Control Manager***: This component manages control flow through all other components of the *CollabAssure* service. All other components work under the same instance of Central Control Manager for synchronous operation. The actions performed are: *i*) Uses network interface manager and participates in ad-hoc market, *ii*) Coordinates with auction manager for bid generation, *iii*) In event of a trade, triggers data download and updates cache, and *iv*) On completion reports the trade and cost to a central entity.

5) ***Download Manager***: The primary job of this unit is to initiate a data download as and when required by the central control manager. The central control manager calls this internal service with *Application ID*, and this unit uses reflections to trigger data download through the application. This data is then returned to the central control manager.

6) ***Security Manager***: To ensure bid confidentiality, *CollabAssure* employs public key encryption scheme for each device pairs involved in a trade. As each device is registered with its respective vendor's market, public keys of the participating devices are stored in the vendor's database, which can be used for authentication and in the later bidding process. When a seller sends its bid to a buyer, he encrypts it with the buyer's public key and sends the encrypted bid to the buyer. The buyer then decrypts it with its own private key and obtains the bid. Since the public key algorithms are known to be computationally expensive, once the buyer identifies a potential seller, a much faster symmetric algorithm will be used to encrypt and decrypt subsequent messages exchanged between the buyer and seller. Specifically, the buyer generates a shared secret key (session key), encrypts it with the seller's public key and sends the encrypted session key to the seller. The seller then uses its private key to decrypt the session key.

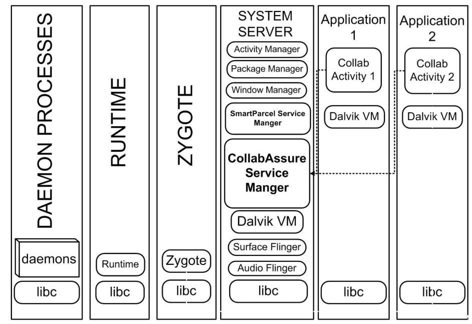


Figure 4: CollabAssure Framework in Android Architecture

7) ***Pricing and Reporting Unit***: As pricing in decentralized ad-hoc networks is a difficult task, we over come this by using two-way reporting strategy where both the devices participating in the trade report to a central entity. Every device maintains a static log of the transactions which may be reported later. This reduces the additional reporting overhead for the buyers and compensates the unavailability of Internet to the buyer devices. For successful payment both buyer and seller must report the same trade. The central entity could be a third party server, but to avert the privacy issues we propose an e-commerce model where each device reports to the operation system (OS) vendors, digital markets [13]. As mobile OS vendors uniquely identify each device, this approach is easy to integrate and maintain.

B. Android and CollabAssure

To amalgamate *CollabAssure* in the Android framework and to be used in Android application development, we modified the android permission model to include the permission named `COLLABASSURE` which uses the underlying network technologies, i.e., BlueTooth, WiFi and NFC for ad-hoc network connections. To use this service android application developers could use the permissions by including the tag "`android.permission.COLLABASSURE`" in the *AndroidManifest.xml*. The service is integrated in the "System Server" module which is launched by Zygote[3] as shown in Fig. 4. While performing the boot operation the Zygote forks the *CollabAssure* service as a system service. This ensures that the *CollabAssure* service gets the system level privileges and is independent of the application context. We modify the android SDK to offer a wrapper around android Activity as *CollabActivity* which provides a raw interface to android application developers offering an additional static update function. This allows *CollabAssure* service to invoke the the update method in the background for each registered application as and when required. We also encourage application developers to use CollabAssure APIs to update the cache when a user himself uses the application, so the cache is always up-to date.

C. Data Security and Privacy

As *CollabAssure* allows only application based data sharing, the system offers complete flexibility for application developers to enforce data encryption at the application level. Typically, *CollabAssure* system targets applications which do not deal with user sensitive data, namely banking, emails, social networks etc. In this scenario, developers of these applications shall not use *CollabAssure* service whereas, news, blogs, RSS feeds etc. benefit from the *CollabAssure* approach. Application developers can also deploy two phase data labeling, i.e., public and private. The public data can be made available for sharing through *CollabAssure*, whereas private and user specific data is not. As an example, a banking application shall mark the nearby ATM locations as public data which can be shared through *CollabAssure* and rest as private data which cannot be shared.

VI. Simulation Setup and Results

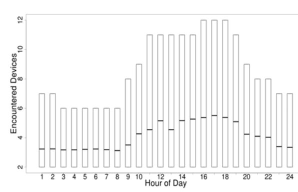
A. Setup

To evaluate the *CollabAssure* ad-hoc market approach we conduct trace based simulations on the *MIT Reality Mining* dataset, containing over 500,000 hours of data collected over a course of 9 months by 100 unique devices [16]. The dataset includes information about call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging and idle) etc. We only focus on the Bluetooh interactions between the users in the data set. Fig. 5 highlights the key characteristics of the dataset. From Fig. 5(a) it is evident that user activity is higher during the day (8:00am - 8:00pm) and lower in the night and early morning hours. Fig.5(b) shows the frequency of devices active per day over the trace period. Device activity in the data set exhibits a minimum of 4 and maximum of 901 devices active per day with mean and standard deviation of 243 and 133 respectively. The distribution of number of devices encountered per scan is shown in Fig. 5(c). Minimum of 2 and maximum of 65 devices per encounter were recorded, overall averaging to 4 devices per encounter and exhibiting standard deviation of 8.67, with only a few occurences more than 20,however.

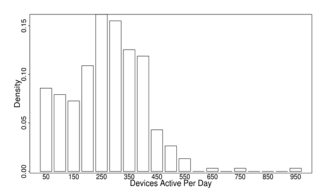
Using this datasets we formulate a set of experiments to analyze the performance of *CollabAssure* using the different parameters such as:

- Data Request Rate (DRR) - The frequency with which the data is being requested by any user.
- Seller Participation Probability (SPP) - The probability of user's participation as seller in a trade, i.e., the probability of user subletting his data.
- Buyer Participation Probability (BPP) - The probability of user's participation as buyer, i.e., whether the user is requesting service and is willing to participate in the auction.

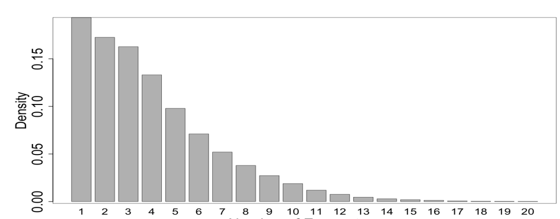
To generate fair and effective bid b_i, we take into account the available data limit for each user and hour of the day into account. Considering a higher mobility and utility of device during the day we pick different values of λ in bid decision. As shown in Eq (3), for higher utility between 09:00 am - 09:00 pm, we randomly pick λ between $[0.4, 0.7]$ and $[0.1, 0.3]$ for night and early morning hours, i.e., 09:00 pm - 09:00 am. We also assign a fixed data limit of 2 GB per billing cycle to each seller and randomly choose a threshold (t_o) between 30% - 80% of the total data limit. For each trade we randomly select data size between 50 KB -150 KB, which is then deducted from each seller's data limit. We measure the utility earned by users during various times of the day. We also analyze the effects of user participation level in both selling and buying roles and make observations about the overall market utility earned on monthly basis.



(a) Hourly Variation of Device Encounters.



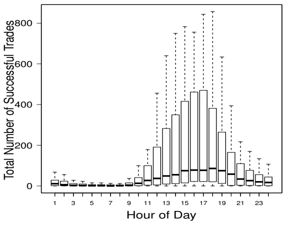
(b) Distribution of Active Devices Per Day.



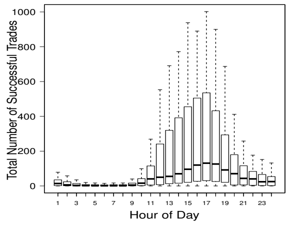
(c) Distribution of Device Encounters.

Figure 5: Variation of User and Device Activity.

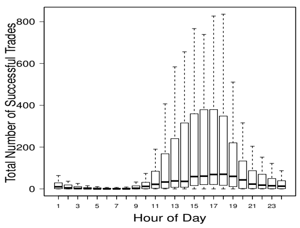
[3]*Zygote* : Zygote is a process which starts at boot time and is the parent of all Dalvik VMs in the system.



(a) DRR = 1 Hour

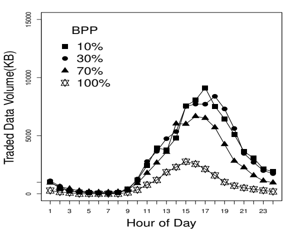


(b) DRR = 4 Hour

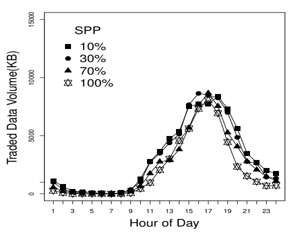


(c) DRR = 12 Hour

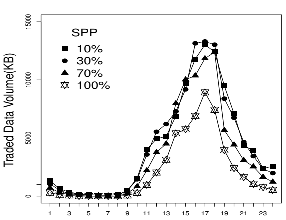
Figure 6: Variation of number of trades with Data refresh rate



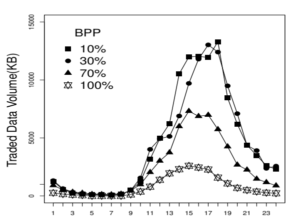
(a) DRR = 1 Hour, SPP=100%



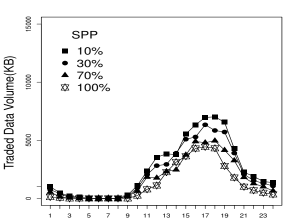
(b) DRR = 1 Hour, BPP=100%



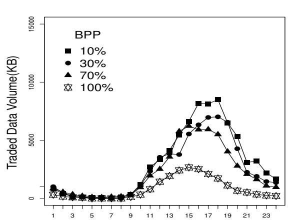
(c) DRR = 4 Hour, SPP=100%



(d) DRR = 4 Hour, BPP=100%

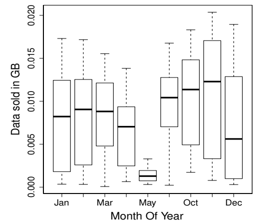


(e) DRR = 12 Hour, SPP=100%

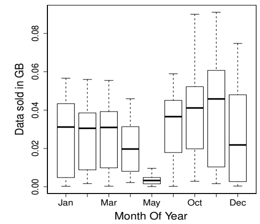


(f) DRR = 12 Hour, BPP=100%

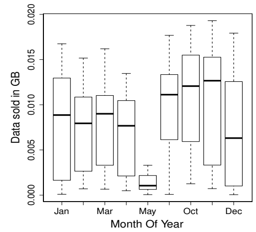
Figure 7: Hourly variation of utility with SPP and BPP



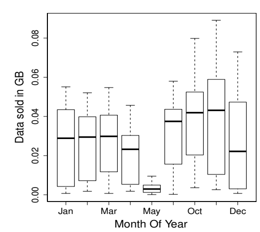
(a) SPP=10%, BPP=10% and DRR=1 Hour



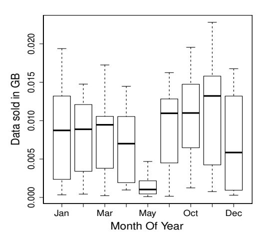
(b) SPP=10%, BPP=30% and DRR=1 Hour



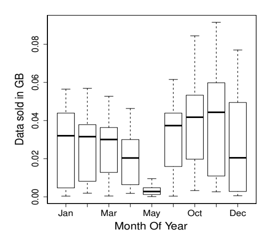
(c) SPP=10%, BPP=10% and DRR=4 Hour



(d) SPP=10%, BPP=30% and DRR=4 Hour



(e) SPP=10%, BPP=10% and DRR=12 Hour



(f) SPP=10%, BPP=30% and DRR=12 Hour

Figure 8: Variation of monthly utility with SPP, BPP and DRR

B. Results

We first analyse the effect of Data Refresh rates (DRR) on the number of trades during the day. From Fig. 5, we observe a higher user activity between 8:00 am to 8:00 pm. We also observe a similar trend with the number of trades. Fig. 6 highlights the successful number of trades at different hours of the day when DRR is varied as 1 hour, 4 hour and 12 hour. From Fig. 6(a) and (c), we observe similar trends in number of trades when DRR = 1 hour and 12 hour respectively. In both the cases, when DRR = 1 hour and 12 hour, a maximum of 800 trades is observed, and median ranging between 0 to 80 trades. Whereas, for DRR = 4 hour we observe a maximum of 1000 trades and a median of 0 to 150 trades per hour.

In our next experiment we focus on the effect of user participation probabilities on the average earned utility. As the market has both buyers and sellers we individually analyze the effect of their participations. The observed results are illustrated in Fig. 7. We first fix the seller participation probability (SPP) to 100% and measure the hourly variation of amount of data sold in each transaction on daily basis when Data Refresh Rates are varied from 1 hour, 4 hour and 12 hour. From Fig. 7(d) we observe that for DRR = 4 hour, the total utility earned is the least when BPP = 10% and is maximum when BPP = 100%, although the trend for the data sold with BPP = 70% and 100% are very similar. From the Fig.7(d), it is apparent that the average data sold during the day time hours is higher and as the number of trades is higher when DRR = 4 hour, the average utility of the market is maximum. Now when we fix the buyer's participation probability BPP = 100% we observe that the average data sold with different seller probabilities show a similar trend. From Fig. 7(b) and (e), it is apparent that sellers participation probabilities has a marginal effect on the average data sold. This trend is in the congruence with the fact that, as user requires data more frequently, there will be an equivalent increase in the data trading. From Fig. 7(c), we observe that SPP is not a major governing factor in deciding the total data utility of the market. When DRR is set to 4 hour, the utility curves for SPP = 30%, 70% and 100% follow a similar trend whereas, this is not observed when DRR = 1 hour and 4 hour respectively.

Next we analyze the variation of overall utility of the market by varying SPP and BPP as 10% and 30% for different data refresh rates. From the results in Fig. 8, we observe that a higher market utility is achieved with increasing buyer's participation. Though, the seller's participation only has a marginal effect on the overall utility. We also observe that maximum utility in the market is achieved when the data refresh rate is 4 hour. The refresh rates, DRR = 1 hour and 12 hour show a similar behavior in earned monthly data utilities. Here it is worth noting that only a partial data is available for the month of May, hence the achieved market utility is comparatively lower than the other months.

VII. Conclusion

In this paper we introduced an ad-hoc market approach to address the issue of data availability to mobile Internet users. We presented the design and implementation of *CollabAssure*, service assurance system for Android devices. The modular architecture of *CollabAssure* allows it to be integrated in any other mobile operating system. The system allows spatio-temporally co-existing mobile devices to participate in an auction for requesting services from neighboring devices. We also discussed the effectiveness of the approach by simulation based experiments. In the future we intend to investigate other auction strategies which can be used in similar setting and optimize the design for a higher battery performance.

Acknowledgement

This research is supported in part by the National Science Foundation under grants CNS-0953620, 0958477, 1116644 and CNS-0845500.

References

[1] B. Kaushik, H. Zhang, X. Fu, B. Liu, and J. Wang, "Smartparcel: A collaborative data sharing framework for mobile operating systems," in *Distributed Computing Systems Workshops (ICDCSW), 2013 33rd International Conference on*, July 2013.

[2] S. Milgram, *The individual in a social world: essays and experiments*. Addison-Wesley Publishing Company, 1977.

[3] G. Hardin, "The tragedy of the commons," *Journal of Natural Resources Policy Research*, vol. 1, no. 3, pp. 243–253, 2009.

[4] "Mobile data offloading for 3g and lte networks," http://www.alvarion.com/applications/mobile-data-offloading.

[5] "Architecture for mobile data offload over wi-fi access networks," http://www.cisco.com/en/US/solutions/collateral/ns341/ns524/ns673/white_paper_c11-701018.html.

[6] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 5, pp. 821–834, 2012.

[7] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1.

[8] B. Braden, D. Clark, and S. Shenker, "Integrated service in the internet architecture: an overview," 1994.

[9] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, "Tycoon: An implementation of a distributed, market-based resource allocation system," *Multiagent and Grid Systems*, vol. 1, no. 3, pp. 169–182, 2005.

[10] S. Pal, S. R. Kundu, M. Chatterjee, and S. K. Das, "Combinatorial reverse auction based scheduling in multi-rate wireless systems," *Computers, IEEE Transactions on*, vol. 56, no. 10, pp. 1329–1341, 2007.

[11] P. Pałka and R. Schoeneich, "Multi-commodity trade application to the routing algorithm for the delay and disruptive tolerant networks," *New Trends in Databases and Information Systems*, pp. 241–250.

[12] "Apple App Store," www.apple.com/iphone/from-the-app-store/.

[13] "Google Play," play.google.com/store.

[14] G. Iosifidis and I. Koutsopoulos, "Double auction mechanisms for resource allocation in autonomous networks," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 1, pp. 95–102, 2010.

[15] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 27–34.

[16] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.