ISSN:0975-9646

Recursive Binary Time Partitioning For Low-Power Mobile Discovery and Bandwidth Allocation in Cloud-Based Wireless Live-Streaming

Mrs. Kaushlya Salunkhe
Computer Engineering
JSPM's ICOER,
Wagholi, Pune

Prof. Mr. Gadekar Devendra
Computer Engineering
JSPM's ICOER,
Wagholi, Pune

***Abstract*--- Now a day's increases the WiFi and Bluetooth capability with increasing prevalence of mobile wireless devices. New applications are emerging that can make use of limited contact opportunities when the devices are physically close. Discovering such services is a challenging problem. It is important to rapidly discover such mobile services to make use of limited contact opportunities. It is also important that to support such applications, we need to design a localized discovery scheme. This scheme can minimize the expected contact latency between mobile phones with limited energy budgets. Number of neighbor discovery schemes existing that, assume lack of any time synchronization. In practice sufficiently accurate time synchronization can be achieved with existing time synchronization techniques. New scheme is introduced Recursive Binary Time Partitioning (RBTP), that determines how the devices should wake up and sleep to achieve minimal contact latency with other nearby devices. Multimedia social networks have been introduced a new technology to improve people's lives through enhanced multimedia distribution. A media cloud system can used to multimedia processing and storage, and provide heterogeneous multimedia services to minimize calculation and reduce time. However, the challenges still remain for end users (e.g., mobile devices and PCs) to receive multimedia streaming from the cloud system with satisfied quality-of-service (QoS). An efficient multimedia distribution approach is used to address these challenges. Taking advantage of live-streaming social networks is innovated in this report to deliver the media services from the cloud to both desktop and wireless end users. Our scheme allows mobile users having limited bandwidth to acquire live multimedia streaming from desktop users.**

Keywords- RBTP, NTP, wake up, sleep, contact latency, multimedia streaming.

1. INTRODUCTION

Today, typical applications and usage patterns of smartphones do not require using the WiFi interface that often. However, to support applications that can exploit the results of neighbor discovery, the phones will need to probe their surroundings at a certain minimum rate, which necessitates the design of improved techniques for managing energy consumption of the WiFi interface. Thus, smarter mechanisms for battery usage are needed. Mobile service discovery on smartphones is challenging due to multiple reasons. Users would not like energy expended by the neighbor discovery service. This service can affect the battery availability for other primary tasks of the phone, such as making or receiving a phone call. Also, the energy budget itself may vary between users, and for each user it may vary with time due to other uses of the device. This makes it challenging to simultaneously optimize the operation of all nodes, especially because the pairs of nodes that will come in contact in the future is not known a priori. And, user mobility is difficult to model and predict the contact patterns, i.e., the time and duration of contacts, are not known beforehand.

The proposed scheme, called Recursive Binary Time Partitioning (RBTP), is a synchronized protocol. It is comprised of the number and the patterns of wake ups for the phones. The number of wake-up instances within a time period, depends on the phone's own energy budget. RBTP wakes up the network interface at certain times and puts it to sleep until the next wake-up instance. In smartphone, the WiFi chipsets already have implemented the functions to quickly switch between wake-up and sleep modes. For example, protocol power saving mode in the 802.11, the mobile device can choose its sleep period in multiples of beacon intervals, and can wake up at the beginning of a beacon interval for a fixed duration of time, which is called the Announcement Traffic Indication Messages (ATIM) window [10]. The beacon interval is longer than ATIM window a Gast [11].

Wireless mobile users also need to pay for multimedia streaming downloading. Thus, this is not recommendable from both content providers' and wireless users' standpoints. They present wireless live-streaming social networks (WLSNs) to deal with the problem. Specifically, the networks allow the desktop users who are watching the same live program to share their live-streaming with the social related wireless users around them through ad hoc wireless communications.

As the number of mobile users tends to be higher than that of desktop users, they need an efficient bandwidth allocation mechanism to coordinate the transmission between the mobile users and their desktop friends.

2. MOTIVATION

Researchers have proposed physical proximity-based applications, such as BlueAware [7], E-SmallTalker [11], and E-Shadow [12]. Eagle and Pentland [7] designed an application that senses users' proximity via Bluetooth. It records the Bluetooth identity of nearby users and queries a database to get their on-file profiles. E-SmallTalker and E-Shadow are similar applications. Teng et al. [26] designed an application that can publish news and messages to users in physical proximity. These applications exploit the

capabilities of the proximity network to establish contact and share messages with nearby devices. Bluetooth is used in [7], and both Bluetooth and WiFi are used in [14]. The energy consumption rate of Bluetooth is lower than WiFi, but the communication range is only about 10 meters that limits its capability. WiFi offers longer distance; however, it consumes more energy than Bluetooth.

2.1 LIVE-STREAMING SOCIAL NETWORKS

Live-streaming social networks are a kind of multimedia social network in which people can exchange multimedia information such as digital video, audio, and images through handheld devices. Multimedia social networks are becoming an emerging research area and also have been the subject of many recent studies. In the study, Lin et al. found that full cooperation between users in peer-to-peer (P2P) multimedia social networks cannot be guaranteed, and some users may even behave dishonestly or maliciously[13]. Then, the authors modeled users' behavior as a repeated game and proposed some cheat-proof and attack-resistant strategies based on incentive to motivate user cooperation. They also addressed security and copyright problems in [14], [15] and [4]. To balance the workload among network nodes, a multimedia social network is created over an overlay topology by following a cross-layer approach that jointly considers characteristics of the overlay at the application layer and schedulability of flows at the medium access control layer. It is found [16] that YouTube has a strong clustering property. Based on this property, a peer-to-peer video sharing social network, NetTube, was proposed aiming at replacing the traditional client/server architecture so as to reduce YouTube server load.

The aforementioned research work focuses on the studies of a variety of multimedia social networks from network design, user behavior analysis, and reducing network loads. However, these studies are mainly for addressing the issues in either a fully online social network or a fully mobile network. The social connections between desktop users and wireless mobile users are not exploited in the previous studies to share the media streaming among mobile and desktop users to improve multimedia QoS. In addition, the efficient resource (e.g., bandwidth) sharing model under a cloud-based infrastructure has not been well studied in the literature.

3. MATHEMATICAL MODEL

3.1 MODEL DESCRIPTION

In the rest of the paper, we will use the term node in place of user, mobile device or a smartphone. The set of nodes is denoted by U. The contact range of a pair of nodes is the istance within which the two nodes can receive each other's beacons. Note that this definition of contact range does not require all nodes to use the same contact range. Time is partitioned into frames of length T, which contains multiples of slots. The slot size is ∂ .We assume that the time difference between clocks on any two devices is bounded by the slot size ∂ . The number of slots in a frame is N=T/ ∂ . In each wake-up instance, a node wakes up for one slot of duration ∂ . It sends a beacon at the beginning and at the end of the slot so that two nodes can discover each other when their slots are overlapping but are not necessarily perfectly aligned. We assume that the beacons are reliable for the ease of description of our solution, but consider the impact of beacon losses in our evaluation. The model of discovery is similar to the slotted time model considered in other works on neighbor discovery [6], [13]. T and ∂ are fixed global parameters across all the nodes. A node's duty cycle d is defined as the fraction of time that a phone remains awake within a frame.

We use the same definition of duty cycle as in previous works [6], [13], [3]. If a node u wakes up n times in a frame, its duty cycle is given by

$$d=n \partial /T \quad (1)$$

3.2 PROBLEM STATEMENT

Assume ui is an arbitrary node in U with a budget of ni wake-up instances in each time frame. ui has no knowledge of the number of wake-up slots of the other nodes. The key question is—What is the optimal strategy that can minimize the average contact latency between any two nodes?

3.3 THE OPTIMUM SOLUTION

Within a frame, if two nodes ui and uj have ni and nj wakeup instances, respectively, then they have at most $n = \min(n_i, n_i)$ wake-up instances to meet. We obtain the optimum pattern as stated below in Theorem 3.1. Observe that a similar result was shown in [30], although their objective of minimizing the missing rate is slightly different from our objective of minimizing the contact latency. Later, we will show that this optimal scheme requires knowledge of the energy budget of the other nodes, and hence is not practically useful.

Theorem 3.1. If n is known to two nodes with synchronized clock, and if the time they arrive into their mutual contact range is uniformly distributed within a frame, then uniform separation of the wake-up instances minimizes the contact latency.

Proof. We consider a single time-frame that begins at time t_0 and ends at time $t_0 + T$. Two nodes u_i and u_j will arrive into their mutual contact range in this time frame. We first investigate an arbitrary scheme that allows u_i and u_j to have $n = \min(n_i, n_i)$ common wake-up instances, which is the maximum number of wake-up instances they could have in common. We assume these wake-up instances begin at time $t_1, t_2, \ldots, t_n$ and they are well separated so that the previous wake up can finish before the next wake-up instance begins. Without loss of generality, we set $t_n = t_0 + T$. The durations between these instances are $I_1 = t_1 - t_0,\ I_2 = t_2 - t_1,\ I_3 = t_3 - t_2, \ldots, I_n = t_n - t_{n-1}$.

The probability that u_i and u_j arrive into the mutual contact range in the interval $[t_{k-1}, t_k]$ is proportional to I_k, where $k \in \{1, 2, \ldots, n\}$. This probability is given by I_k / T. If ui and uj arrive into their mutual contact range in interval $[t_{k-1}, t_k]$ the expected contact latency is Ik=2. So, the expected contact latency ID_{exp} within this frame is

$$\mathbb{D}_{exp} = \sum_{k=1}^{n} \frac{I_k}{T} \times \frac{I_k}{2}. \quad (2$$

Using $\sum_{k=1}^{n} I_k = T$ and the Cauchy-Schwarz inequality, we can get $\mathbb{D}_{exp} \geq T/2n$ when $I_1 = I_2 = \cdots = I_n$, $\mathbb{D}_{exp} = T/2n$ which is the optimal expected contact latency represented as follows:

$$\mathbb{D}_{exp}^{\mathrm{OPT}} = \frac{T}{2n}. \quad (3)$$

In the worst case, two nodes will move to their mutual contact range at the beginning of the largest interval and then they have to wait until the end of that interval to discover each other. We denote the worst-case contact latency as ID_{worst}:

$$\mathbb{D}_{worst} = \max_{k=1,\ldots,n} \{ I_k \}. \quad (4)$$

Observe that the pattern described in Theorem 2.3.1 also minimizes ID_{worst}. The worst-case contact latency in the optimal scheme is

$$\mathbb{D}_{worst}^{\mathrm{OPT}} = \frac{T}{n}. \quad (5)$$

According to the model in Section 5.1, $T/n = \delta/d$. The above analysis indicates that the frame size does not affect $\mathbb{D}_{exp}^{\mathrm{OPT}}$ and $\mathbb{D}_{worst}^{\mathrm{OPT}}$ if the duty cycle d and the slot size ∂ are fixed. In the remaining paper, we fix the frame size to be *N = 1,024* slots, thus $T = 1{,}024\delta$.

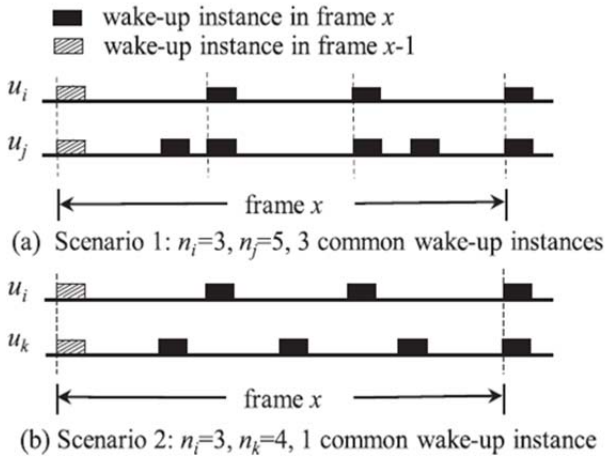

Figure3.1 (a) both u i and u j have n i and n j and their wake-up patterns a priori. (b) u i and u k have no a priori knowledge of each other's number or pattern of wake-up instances. The dashed vertical lines indicate the common wake-up instances of the two nodes.

Figure 3.1 shows an example, where the minimum number of wake-up instances between the two nodes is 3 . If two nodes know that they are going to meet in the future, but do not know the time when they will meet, then they can determine a pattern of wake up that minimizes the expected contact latency. Such a pattern is based on Theorem 3.1 and is shown in Figure 3.1a. However, this is impractical to implement as it requires prior knowledge of which node pairs are going to meet next and their energy budgets. Without such knowledge, the scheme will not remain optimum.

For example, in Figure 3.1b when u i comes in range of another node u k with the energy budget of four wake-up instances, but with a different wake-up pattern than u j , the optimality of contact latency is lost. From this example, we observe that a good scheduling strategy should try to make two nodes meet for the maximum possible number of instances, i.e., $\min(n_i, n_j)$, where the wake-up instances are well separated within the frame. In the next section, we will present a localized scheme that achieves near-optimum performance both in terms of the expected and the worst-case contact latencies.

3.4 MULTIMEDIA CLOUD

Multimedia social networks are characterized by a large number of multimedia files with big size. The storage of this multimedia content is shown to be a significant challenge. This undesirable situation can be leveraged by making use of the method of infrastructure as a service (IaaS) in cloud computing. Many organizations can provide commercial clouds that offer access to virtualized resources (storage, computation, and application). With the virtualized resources, the content provider can build a multimedia cloud (Figure 3.2) that contains two main components

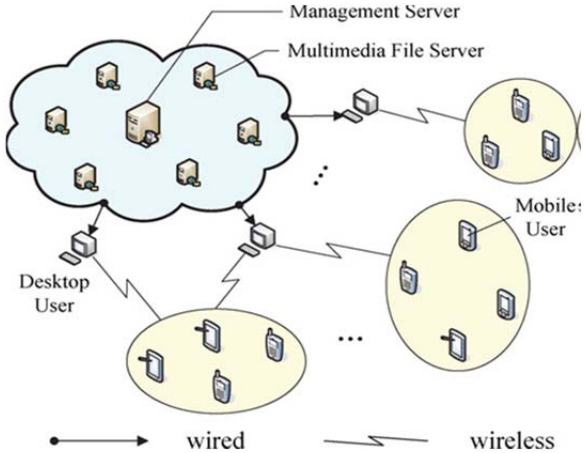

Figure 3.2 Multimedia clouds for wireless live-streaming networks

1. Multimedia file server: this server primarily provides the service of multimedia files storage for the content provider, and delivers the content to the end users in the network. Multimedia processing and file classification are also within its capability.
2. Management server: it functions as a central controller that schedules the distribution of enormous multimedia files among, and allocates tasks or requests from the end users.

More specifically, the content provider can distribute the multimedia files to a bunch of servers which may be placed

at different locations. The users who require different programs can request the corresponding file servers. The multimedia file of a program can be divided into a number of clips and distributed to several servers. In this way, the working load of the servers can be balanced. Note that their requests are first reported to the head (i.e., management server) of the multimedia cloud. The head has the information of all saved files in a specific file server. Then, the requests are distributed to the corresponding server by the head. Upon receiving the requests, the server transmits relevant files to the end users with high speed.

3.5 BANDWIDTH ALLOCATION PROBLEM

They consider a WLSN consisting of a set *M* of desktop users and a set *N* of mobile users, *M ≡ {1, 2, · · · ,M} and N ≡ {1, 2, · · · ,N}*. Some mobile users in this network may have common social properties, such as location, profession, interest, education, hobby and so on. In this sense, they can group these mobile users according to their social contexts. They suppose that these *N* mobile users are divided into a set *G* of groups, *G ≡ {1, 2, · · · ,G}*. Users in the same group can communicate with each other freely, and private information can be exchanged among them. Furthermore, members in the same group can connect to distinct desktop friends if available, and each mobile user can connect to only one desktop user at a time.

In this WLSN, each desktop user is willing to share its live-streaming programs by allocating portion of wireless bandwidth to its mobile friends. Let *bi* denote the size of bandwidth that desktop user *i* is willing to share. Generally, *bi* is bounded, i.e., $0 \le bi \le \bar{}bi$ $(i \in M)$ where $\bar{}bi$ is the maximum value of bandwidth that desktop user *i* can share. Let *ni* denote the total number of mobile users connected to desktop user *i*. Without loss of generality, the mobile users connected to the same desktop user *i* are allocated the identical size of bandwidth, which is expressed as *bi/ni*. Let *ng i* be the number of mobile users in group *g* who choose desktop user *i*. Then, they have *ni* = _ *g ng i* . Thus, the size of bandwidth for each mobile user in group *g* is given by *bi/ g ng i* .

For each mobile user, its strategy set is the set of desktop users to which this mobile user can connect to. Intuitively, in order to experience high-quality live video, each mobile user wishes to watch the live programs shared by the desktop friends with large size of allocated bandwidth. However, this may also lead to congestion in this specific desktop user if it is simultaneously connected by a large number of mobile users. To address the problem, a pricing scheme is presented here for the desktop users to dynamically adjust their load. The shared live-streaming programs are not free. Mobile users need to provide some kind of payment. They suppose that the mobile users connected to the same desktop user *i* are charged an identical price, which is denoted by *pi*. It can be seen that there is an inherent tradeoff between bandwidth and price for each desktop user. Once all desktop users have made their individual decisions, their chosen strategies form a profile ***(b, p)***, where ***b*** *= [b1, b2, · · · , bM]T and* ***p*** *= [p1, p2, · · · , pM]T* . The strategies profile for all desktop users except user *i* is denoted by ***(b−i, p−i)***. Then, ***(b, p)*** *= (bi,* ***b****−i; pi,* ***p****−i).*

4. DESIGN PROCESS

4.1 RECURSIVE BINARY TIME PARTITION

The main idea of RBTP is that a node wakes up at instances that recursively partition a frame in a binary fashion. If a node has *n = 1* wake-up instance in a frame, it wakes up at the end of the frame. When a node has *n > 1* wake-up instances, it determines its wake-up time by calculating x and m using

$$n = 2^x + m,\ 0 \le m < 2^x, x \in \mathbb{N}, m \in \mathbb{N}. \quad (6)$$

Then, it partitions the frame into 2x equal-length intervals with 2x wake-up instances. RBTP places the remaining m wake-up instances in the middle of the first m intervals for the sake of simplicity. With this schedule, denote t_k as the start time of the *k*th *(k > 0)* wake-up instance in the frame. When *n > 1*, t_k is calculated by (7), where t_0 is the starting time of the frame and T is the length of a frame. The node will switch to sleep mode at time $t_k + \delta$:

$$t_k = \begin{cases} t_0 + kT/2^{x+1}, & k \in \{1, 2, \ldots, 2m\} \\ t_0 + (k-m)T/2^x, & k \in \{2m+1,\ 2m+2,\ \ldots, n\}. \end{cases}$$

4.2 SMARTPHONE CLOCK SYNCHRONIZATION

In this section, we show that clock synchronization can be achieved with small energy overhead when using TP. In theory, the clock can be synchronized using GSM, 3G or 4G protocols. However, these protocols are provided in proprietary binary modules and do not have an interface for application developers to synchronize the smartphone's clock. In contrast, NTP is more convenient to implement on the Android phones. We have implemented the linear fitting method and the linear programming method [18] based on NTP to calibrate the clocks.

The terminology used here to describe a clock is similar to [17]. A clock on a node i is represented by C_i. A true" clock is a clock that always returns the "true time." Clock C_i returns time $C_i(t)$ at the true time t. The offset of clock C_i is defined as $C_i(t) - t$, i.e., the difference between clock C_i and the true time t. The frequency of clock C_i is defined as $C_i'(t)$, i.e., the rate at which the clock increases. The skew of clock C_i is defined as the difference between $C_i'(t)$and the frequency of the true clock, which is 1 sec/sec.

In our clock synchronization experiment, the smartphones are programmed to get m time stamps from the NTP servers in a synchronization instance. The synchronization instances are equally separated by H hours. Because the round trip delays between the smartphones and the NTP servers vary with time, multiple time stamps are retrieved in each synchronization instance. A phone I logs its local time $C_i(t)$ and the NTP time $C_i^{ntp}(t)$. The linear fitting and the linear programming methods are used

to estimate the skew of a phone based on the logs. Let the vector of the local clock log of a phone be C_i and the corresponding NTP clock vector be $\mathbf{C}_i^{ntp}$. By fitting the linear function $\mathbf{C}_i^{ntp} = \text{skew} \cdot \mathbf{C}_i + b$, we can obtain the estimated skew. Moon et al. [18] provide another method that can find the skew by using linear programming. In our implementation, in each synchronization instance 10 NTP time stamps are retrieved and both $\mathbf{C}_i^{ntp}$ and C_i contain the time stamps from the previous two NTP synchronization instances.

4.3 EVOLUTION PROTOCOL FOR MOBILE USERS

They present an iterative algorithm for the mobile users to converge to the evolutionary equilibrium. Based on the replicator dynamics, each mobile user changes its strategy to maximize its own utility. In order to achieve the evolutionary equilibrium, an evolution protocol is presented for each mobile user. Since the users in the same group can exchange information, the current choice and utility of one mobile user is available to others, but not available to the users in different groups. They assume that a mobile user can also receive the average utility of its own group. The specific procedure of this protocol is described as follows.

1)Initially, each mobile user connects to an available desktop user randomly;

2) Each user computes its utility using the allocated bandwidth and the charged price by the connected desktop user according to (2). The allocated bandwidth is measured by this mobile user himself since the total number of mobile users connected to the same desktop user in (2) is unavailable.

3) After communicating with the other users in the same group, each user gets its data about choice and utility, and then computes the average utility of the group as

$$\bar{\pi}^g(t) = \frac{\sum_i n_i^g(t-\tau) \cdot \pi_i^g(t-\tau)}{n^g(t-\tau)}.$$

4) If the average utility is greater than its own utility, the mobile user changes its connection to another desktop user who offers higher utility with a possibility of

$$\theta(t) = \frac{\bar{\pi}^g(t) - \pi_i^g(t)}{\bar{\pi}^g(t)}.$$

Otherwise, the user keeps the current connection.

5) Repeat procedure 2) to 4).

As indicated by the evolution protocol, all users in the same group can obtain equal utility at the equilibrium.

Algorithm1 Executed by each mobile user

1. Build initial connection
2. repeat
3. Compute utility using the received price and measured bandwidth
4. Get information about the choices and the utility of all other users in the same group
5. Compute the average utility $\bar{\pi}g(t)$
6. if $\bar{\pi}g(t) > \pi g\ i\ (t)$ then
7. Change the connection with probability $\theta(t)$
8. else
9. Maintain the current connection
10. end if
11. Until all mobile users in the same group have equal utility.

Algorithm2 Executed by each desktop user

1. For each desktop user:
2. Set the initial bandwidth willing to share, b_i ,0 and price charged for the bandwidth, p_i ,0
3. repeat
4. Wait a period time T_w for the evolution among mobile users
5. Update the size of bandwidth and the price, set $b_i(t+1)=\max\{\min\{ b_{i0}(t+1), b_i \},0\}$, and set $p_i(t+1)=\max\{ p_{i0} (t+1),0 \}$
6. $t=t+1$,
7. Until b_i and p_i are both unchanged.

4.4 COMPETITION AMONG DESKTOP USERS

Based on the result of the evolutionary game for the mobile users, the desktop users will compete with each other and update their strategies in order to maximize their own utilities. They model the competition among the desktop users as a non-cooperative game, and consider the Nash equilibrium (NE) as the solution to the game. In this non cooperative game, players refer to the desktop users. The strategies are the size of bandwidth that a player is willing to share and the price that a player charges. The utility is defined as the difference between the total payment from the mobile users who connect to this desktop user and the cost for transmitting live-streaming files.

5. CONCLUSION

RBTP has bounded performance with respect to an optimum solution in terms of expected and the worst-case contact latencies. We observed that significant reduction in latency can be achieved by leveraging time synchronization. Our experiment shows that clock synchronization requirement of RBTP is achievable and its energy overhead is negligible. The simulations show that the performance of RBTP is significantly better than state-of the art asynchronous protocols. We also proposed a cloud-based WLSN in which desktop users received multimedia services from a multimedia cloud and they shared their live contents with mobile friends through wireless connections. This network architecture offered advantages of saving the cost for network services and satisfied the increasing demand on bandwidth requirements. In this multimedia social network, we formulated a bandwidth allocation problem with the objective to share bandwidth efficiently for both desktop users and mobile users.

REFERENCES

1. Dong Li , Prasun Sinha "*RBTP: Low-Power Mobile Discovery Protocol through Recursive Binary Time Partitionin" IEEE Transactions on Mobile Computing*, vol. 13, no. 2, february 2014
2. Guofang Nan, Zhifei Mao, Mei Yu, Minqiang Li, Honggang Wang, and Yan Zhang, *"Stackelberg Game for Bandwidth Allocation inCloud-based Wireless Live-streaming Social Networks"* IEEE Systems Journal, vol. 8, no. 1, March 2014
3. G. Anastasi, M. Conti, E. Gregori, and A. Passarella, *"802.11 Power-Saving Mode for Mobile Computing in Wi-Fi Hotspots: Limitations, Enhancements and Open Issues,"* Wireless Networks, vol. 14, no. 6, pp. 745-768, 2008.
4. P. Bahl, R. Chandra, and J. Dunagan, *"SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks,"* Proc. ACM MobiCom, pp. 216-230, 2004.
5. M. Bakht, M. Trower, and R.H. Kravets, *"Searchlight: Won't You Be My Neighbor?"* Proc. ACM MobiCom, pp. 185-196, 2012.
6. F. Ben Abdesslem, A. Phillips, and T. Henderson, *"Less Is More: Energy-Efficient Mobile Sensing with Senseless,"* Proc. First ACM Workshop Networking, Systems, and Applications for Mobile Handhelds, pp. 61-62, 2009.
7. R. Chandra and B.P. Bahl, *"MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single*
8. *Wireless Card,"* Proc. IEEE INFOCOM, vol. 2, pp. 882-893, 2004.
9. P. Dutta and D. Culler, *"Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications,"* Proc. Sixth ACM Conf. Embedded Network Sensor Systems (SenSys '08), pp. 71-84, 2008.
10. N. Eagle and A. Pentland, *"Social Serendipity: Mobilizing Social Software,"* IEEE Pervasive Computing, vol. 4, no. 2, pp. 28-34, Jan.-Mar. 2005.
11. S. Vasudevan et al., *"Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem,"* Proc. ACM MobiCom, pp. 181-192, 2009.
12. HTCG1, http://en.wikipedia.org/wiki/HTC_Dream .com , June 2012.

Mrs. K. B. Salunkhe received his B.E. degree in Computer Engineering from SVPM's College of Engineering University of Pune, Maharashtra, India, in 2006, the M.E. degree appearing in Computer Engineering from JSPM's BSIOTR University of Pune, Maharashtra, India. She was lecturer with Department of Computer Engineering Course, SVPM's Institute of Technology, Baramati, Maharashtra, India from June 2006 To Dec 2007. She was a HOD with department of Computer Engineering (II Shift), JSPM's BSPoly, Pune, Maharashtra, India, From Jan 2008 to Jan 2015. Now She is working assistant professor in JSPM's ICOER University of Pune, Maharashtra, India, His research interests include mobile computing, and cloud computing.