Peer-to-Peer Netw. Appl. (2012) 5:178192

Towards an effective integration of cellular users to the structured Peer-to-Peer network

Mohammad Zulhasnine · Changcheng Huang · Anand Srinivasan

Abstract Integration of cellular users into the peer-to-peer (P2P) network is still in limbo due to limitations caused by heterogeneity, mobility and time-varying capacities of the wireless channel. If traditional Chord is employed to include users from the cellular networks, users under the same base station scatter in logical topology randomly. In this paper, we present a novel cellular Chord (C-Chord) P2P system that integrates the cellular users into the well-established structured P2P network in a topology-aware fashion. C-Chord offers the cellular users a choice of downloading contents either from the Internet peers at a faster rate or from other cellular users from the same base station avoiding the Internet data penalty. We also incorporate the peer selection module based on stable marriage problem that chooses the appropriate candidate from the discovered potential senders. We conduct extensive simulations to evaluate the performance of the proposed C-Chord P2P system and the peer selection module. Simulation results show that the path-length per lookup query is smaller than that of the traditional Chord system. Overhead due to renewal of routing information is also smaller for the cellular users in the C-Chord system. We also measure the throughput at the cellular receivers to analyze the effects of selecting peers either from same base station or from outside the Internet gateway. Throughput also increases dramatically due to an intelligent selection of peers among the potential senders.

M. Zulhasnine and C. Huang
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
E-mail: {mzulhasn,huang}@sce.carleton.ca

A. Srinivasan
EION Inc., Ottawa, Canada
E-mail: anand@eion.com

Keywords peer-to-peer (P2P) · cellular network · structured peer · topology-aware · peer selection · stable-marriage

1 Introduction

Peer-to-peer (P2P) system is exploding with great interest which aims to overcome most of the main limitations of the traditional client/server architecture. P2P systems are classified into unstructured P2P system such as Gnutella [1], KaZaA [2] and structured P2P system such as Chord [22], Pastry [20], content based network (CAN) [19]. Unstructured lookup protocols, in [1], [2], flood the queries with restriction and therefore accrue considerable message overhead and also lessen the search accuracy. Structured lookup protocols in [22], [20], [19], construct overlay network using distributed hash table (DHT) and thereby improve search efficiency and accuracy. Although there remain some differences on how DHT shapes the overlay network and routes search operation in [22], [20], [19]; Chord is the simplest and prevalent protocol.

As more appealing applications and services are offered, we are being less wired on the contrary. Fourth-generation (4G) mobile technology aims at meeting the growing demands of supporting data applications at a faster speed. Cisco estimates 131 percent compound annual growth rate of the mobile traffic between 2008 and 2013 [5]. Cellular users can share ring tones, themes, wallpapers, games, or music by joining into the P2P paradigm rather than downloading from commercial content providers. Integration of cellular users into the P2P network is still in limbo due to limitations caused by heterogeneity, mobility and time-varying capacities of the wireless channel. P2P searching protocol, which relies on overlay network, yields some penalties in terms of bandwidth usage due to a partial removal of routing intelligence of the network layers. Several researchers emphasized on the scalable topology control protocol to discover neighboring wireless peers to save valuable wireless bandwidth [15], [23], [11], [4]. Some of the researchers addressed the wireless related issues in combining wireless mesh network (WMN) [23], [4], [14], [12]. Few authors also studied how to integrate the cellular network into the P2P paradigm [11], [24], [10], [3]. The works in [23], [14], [12] are only suitable for the infrastructure-less wireless network. The methods in [15], [4] require the knowledge of physical layer and therefore are unsuitable for application driven P2P system.

In this paper, we propose an efficient P2P system that integrates cellular users in topology-aware style. The proposed C-Chord P2P system enforces a certain desirable structure to the overlay network. Our proposed system needs base station's unique numeric identification number (Cell-ID), already available to all cellular users, and does not include the base station as one of the nodes on the identifier circle, unlike [24]. Different from [11], our model does not require any super peer that maintains indexes of shared file and peers' location.

Appropriate peer selection from the potential senders also plays a vital role for P2P file sharing over wireless networks [9], [6], [16], [17], [21]. Most of these works select the potential sender(s) that is optimal for one requesting peer. Our stable marriage (SM) and heuristic peer selection module consider the scenario where multiple requesting peers download contents from multiple potential senders. The key contributions to the work are:

- Our proposed C-Chord P2P system integrates the cellular users into the popular Chord P2P system in a topology-aware fashion. Cellular users have the option to choose between Internet peers and cellular peers. The users of the cellular network can either download diverse content from the stable Internet peers at a faster rate or download social contents from the peers within the same base station avoiding the Internet data penalty.
- Our proposed system reduces the high management cost of routing information with fewer entries in the routing table for the cellular peers.
- We measure the mean hop-count per lookup request and compare it with that of the traditional Chord system.
- We measure the throughput experienced by the cellular receivers to analyze the effects of peer selection strategy.
- We formulate the problem of optimal peer selection as SM problem and also integrate a heuristic peer selection module. The results show that SM and heuristic peer selection strategies results in higher aggregate throughput than that of a random peer selection strategy by choosing potential sender with better load distribution.

We organize the remaining part of this paper as follows. In section (2), we review works relevant to our proposal. Section (3) presents the system model and states the problem. Section (4) describes the proposed C-Chord P2P system along with peer selection strategy. Section (5) presents simulation results of our experiments. At the end, we conclude our work in section (6).

2 Related work

P2P over wireless network has received significant attention; a combination of both offers new possibilities, but poses several challenges as well. In this section, we briefly discuss previous works related to our work.

2.1 Topology-awareness in wireless P2P network

One of the major factors that affect the efficiency of the P2P system is the degree of mismatch between the P2P overlay and its underlying physical network. [23] proposed a topology-aware P2P protocol based on Chord for the infrastructure-less wireless network. Entering node broadcasts request to join the existing wireless P2P network. Any node upon receiving the entering application, calculates the relative distance from the energy level of the received signal. From the received energy strength and identifier (*id*) space margin information, the entering node gets a node *id*. Physically close nodes get adjacent *id*s exploiting the nodes' position information. The method is only suitable for the infrastructure-less wireless network and does not consider the Internet gateway in the design. [4] proposed a location-aware Chord based P2P algorithm for the wireless mesh network to match the overlay network and the physical network. However, this method requires location-awareness possibly using the GPS receivers. The users also may not feel comfortable to release the geographical location due to privacy and security reasons. The authors in [14] proposed an application-level-driven topology control protocol for a wireless P2P file sharing network. To construct an adjacency set, the algorithm considers contribution levels of different nodes; the popularity of file resources owned by individual nodes, aggressiveness of the file-requesting node, and the remaining energy levels of nodes. The method is only suitable for the infrastructure-less wireless network and does not consider the Internet gateway in the design.

2.2 Adapting DHT technique for the wireless network

Adaptation of the existing DHT techniques to the wireless networks through topology awareness has received considerable attention recently. The authors in [15] proposed a P2P file sharing algorithm for wireless mesh network. In this algorithm, part of the lookup messages is constrained in a local mesh network, and then directed to DHT based P2P network if required. The results show that this method can greatly reduces the number of lookup messages in physical networks. However, this algorithm requires full awareness of the underground network topologies and also does not mention how the base stations are connected to the DHT based P2P network. In [12] the authors proposed a node *id* based P2P algorithm that minimizes the overhead of updating DHT routing data structure and the stored data as a node joins/leaves the system. The algorithm classifies the participant nodes into two categories: the reliable nodes that are stable and the leaf nodes which join/leave very frequently. The reliable nodes, with higher bits on left side, cover the entire *id* space and play more important role of the routing and the replication. The leaf node gets the node *id* with lower bits on the left side which makes its *id* region as small as possible to minimize the information maintenance overhead for joining/leaving of it. The leaf nodes only assist the reliable nodes. It is not understood how the authors determined the node reliability; the ratio of the reliable nodes and the leaf nodes also affects the scalability.

2.3 Integration of cellular users onto the P2P world

There has been considerable research in integrating cellular network into the P2P system [24], [11], [10] [3]. [11] proposed a network-aware P2P file architecture for wireless mobile networks. This scheme assigns the peers to a network-aware cluster using a network prefix division, and thereby enables the files to be searched first with nearby peers. The bootstrap peer maintains up-to-date cluster routing table to direct new peers joining the network to the appropriate clusters. The authors in [10] investigated the performance benefits and drawbacks of using the P2P network model for Internet access in cellular wireless data networks. They claimed that the improved spatial reuse of the P2P network does not translate into better throughput performance instead might actually degrade the throughput performance of the network. This is due to a bottleneck of the channel around the base station, protocol inefficiencies and host mobility. The authors suggested base station assistance and multi-homed peer relays approaches to achieve the performance enhancement while providing fair service and resilience to mobility. However, support from management and network operator is most unlikely in the context of P2P network. In [3], the authors suggested that an extended peer (non cellular user) from P2P network can communicate with cellular users as a client/server based communication between them. In this way cellular users can participate indirectly in the P2P network, using the extended peers as proxies and also avoid the costly competition for resources. However, in this method the cellular user needs an extended peer within communication range. The authors in [24] proposed an architecture where the base stations form the main Chord and cellular users form auxiliary Chord. Because the nature of P2P applications is user-driven; cellular network operators and P2P users have conflicting preferences and therefore base station's participation in the C-Chord is questionable. Unlike [24], we do not advocate for base station's participation in our architecture rather cellular user only uses base station's Cell-ID. Also the architecture in [24] only considers base stations and cellular users, and does not consider wired users. Whereas, our motivation is to integrate the cellular users to the Internet, and hence we consider wired users in our proposed architecture.

2.4 Importance of appropriate peer selection

Peer selection strategy is one of the major challenges towards an efficient P2P system in the wireless network. In [9], Heffeda et al. proposed a P2P media streaming system which selects best peers through monitoring the status of peer connection and reacting to peer connection's failure or degradation. The simulation results show that topology aware peer selection approach performs better by inferring and leveraging the underlying network conditions. This scheme applies to large-scale P2P networks and does not address the wireless related issues. In [6], the authors addressed the issue of increased contention on the shared wireless channel when multiple nodes try to access the same file simultaneously. They also proposed a cooperative P2P file transfer protocol which increases the aggregate throughput by selecting potential download peers with a minimum interference on the download paths. However, this protocol selects potential download peers based on the current load and interference on the download paths for a single receiver at a time. The authors in [16] also emphasized the importance of appropriate peer selection from the discovered potential sender peers. In their proposed algorithm, the candidates for the senders send an evaluation scores based on several factors such as energy, link quality, movement, lingering time and security. The receiver peer waits a period for the evaluation scores from several candidates before ranking them. However, the evaluation score changes repeatedly in the time-varying wireless environment, and the price paid to acquire evaluation scores is not worthwhile most likely. The work in [17] measured the performance of mobile phone devices in a Chord-based P2P Session Initiation Protocol (P2PSIP) overlay network. The authors concluded that memory, and CPU of the mobile peer sustain without impairments. However, the message overhead in traditional Chord increases the wireless network overload and battery consumption, and thereby degrades the performance of mobile peers. The work in [21] proposed a multiple senders selection scheme to improve the performance in terms of data rate and energy consumption in the context of the multi-hop wireless network. The sender selection process is based on the restless-bandit algorithm where the potential senders' states (active or passive) are determined in a distributed and cooperative manner.

3 System model and the problem statement

In the steady state, there are N wired peers that form the main chord with K keys. There are additional U cellular peers from V number of cell sites that form V auxiliary Chord rings. u_i denotes any cellular peer which belongs to $i-$th base station. U_i denotes the number of cellular peers of $i-$th auxiliary Chord or under the $i-$th base station, that is in the steady state $\sum_{i=1}^{V} U_i = U$.

Peers, in the Chord system, construct an overlay network with location-independent virtual addresses. If traditional Chord is employed to include users from the cellular networks, users under the same base station scatter in logical topology randomly. Cellular users thus do not have the option of choosing between local peers and the Internet peers. Mobile network operator provides the Internet service to the cellular users through the Internet gateway. When a cellular user accesses the Internet; it pays for the Internet usage, and we term it as the Internet data penalty. Since Internet usage price varies depending on location and service-package, we measure the Internet data penalty in terms of bandwidth usages. By choosing local peers (i.e. peers from the same base station) cellular users may not only avoid Internet data penalty but also may look for direct connectivity employing device-to-device (D2D) communication. On the other hand cellular users, with better data plan from the commercial provider, may also like to connect with Internet peers to download diverse content at a faster rate. We therefore need to design an efficient P2P system that integrates cellular users in a topology-aware fashion, and also offers a selection choice between local peers and Internet peers. The design should also reduce (compulsory copies and update) cost for routing information with peers joining/leaving the network all the time.

Any requesting peer in the wireless P2P system may find multiple potential senders. Other requesting peers may seek same file segments and eventually discover same potential sender(s). When multiple requesting peers try to access same potential senders simultaneously, contention may occur on the shared wireless channel. Therefore, we need to incorporate an efficient peer selection module that selects potential sender intelligently.

4 The topology-aware Cellular Chord P2P system

We use the simplest and prevalent protocol Chord as the base protocol. Although we choose Chord because of its simplicity, our design is also applicable to any other DHT based system such as Pastry [20], CAN [19]. First, we briefly describe the base Chord mechanism and then illustrate the details of our proposed C-Chord P2P system.

4.1 Background of the base Chord P2P system

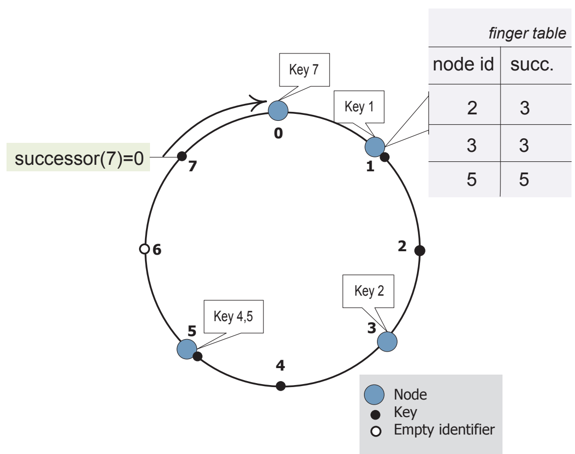

Fig. 1 A 3 bit ($m = 3$) identifier circle with 4 nodes: 0, 1, 3 and 5. For clarity of the picture, we only show the successor of node 7 and the finger table of node 1.

Chord provides a mechanism to store key/data pair onto nodes responsible for them in a distributive manner employing consisting hashing function and efficiently locates the node that stores a specific data item associated with a key. A key k is associated to each resource available in the network. An m bit number or *id* is generated for each key k by hashing the resource name, and for each node n by hashing the IP address. Chord maps both peer and key *ids* on the same circle of numbers from 0 to $2^m - 1$. We use the notation id_n and id_k just to indicate that the *ids* belong to a node n and a key k, respectively. Each key is stored to the first node whose *id* is equal to or follows (the *id* of) k in the circle. This node is defined as the *successor node* of key k and denoted as $successor(k)$. Fig. 1 depicts a 3 bit ($m = 3$) *id* circle with 4 nodes: 0, 1, 3 and 5. For an example, the successor of *id* 7 is 0. Therefore key 7 is located at node 0. Similarly, key 1 is located at node 1, key 2 at node 3 and key 4, 5 at node 5. To accelerate the lookup process, each node n maintains a routing table with up to m entries. This routing table is commonly known as the *finger table*. The i-th entry in the table of node n, contains an *id* id^i and its successor s_i. id^i succeeds n by at least 2^{i-1} on the *id* circle, and therefore $id^i = n + 2^{i-1}$ and $s_i = successor(id^i)$; where all arithmetic is modulo 2^m and $1 \leq i \leq m$. For instance in Fig. 1, the *finger table* of node 1 contains the successor nodes of *ids* $(1 + 2^0) \bmod 2^3 = 2$, $(1 + 2^1) \bmod 2^3 = 3$, and $(1 + 2^2) \bmod 2^3 = 5$. Readers are encouraged to read [22] for detail descriptions. Although routing in Chord becomes more efficient due to the introduction of the *finger table*; maintenance overhead of the DHT routing structure grows with the number of entries in the *finger table* and the peers' churn rate.

4.2 The proposed C-Chord P2P system

The proposed C-Chord P2P system incorporates the cellular network into the traditional large-scale Chord model where the cellular users under the same base station may choose to communicate locally or harvest diverse contents from the Internet peers as well. C-Chord model consists of one large main Chord (m-Chord) ring that contains Internet peers, and several (equal to the number of base stations) auxiliary Chord (a-Chord) rings containing all the cellular peers. Fig. 2 shows the relationship between physical and logical networks in the proposed C-Chord system. Here, the m-Chord consists of 4 wired peers and each a-Chord consists of two cellular peers. The cellular peers under the same base station belong to a particular a-Chord. The base station does not participate in any of the Chord. Fig. 3 illustrates the overlay network in details. a-Chord rings

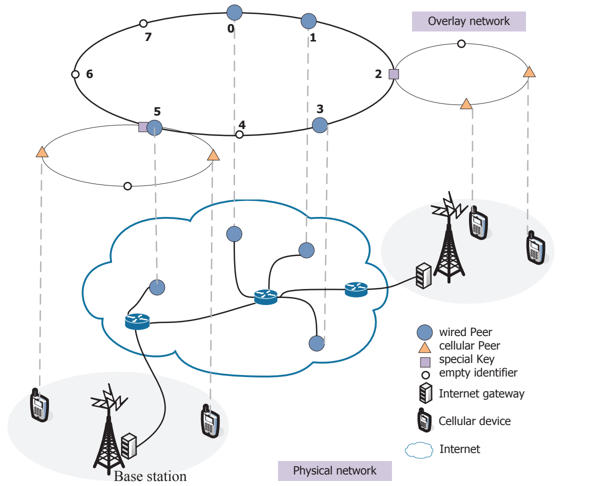

Fig. 2 The relationship between physical and logical networks in the proposed C-Chord system. For the clarity of the picture; only the special key and part of the network are shown.

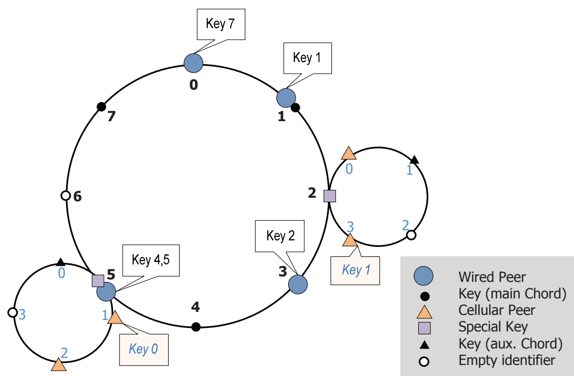

Fig. 3 The C-Chord P2P overlay network.

are attached to the m-Chord ring exploiting a *special key*. For instance in Fig. 3, each a-Chord is connected to the m-Chord employing special keys 2 and 5. We describe this special key in the next paragraph. Cellular users with the same base station are part of the same a-Chord ring and therefore, discover the neighboring wireless peer through application driven mechanism which is more realistic in the context of P2P file sharing.

We design the *special key* and the data associating it such that each a-Chord is tightly coupled with the m-Chord. Every base station has a unique numeric identification number (Cell-ID) and cellular users retrieve this Cell-ID through cell search procedure. The unique Cell-ID serves as the common key to the users under that particular base station. This is the beauty of our proposed mechanism. An m bit *id* is generated by hashing each Cell-ID and denoted as id_{cell}. The data associates with this special key with identifier id_{cell} has the format $\mathcal{IP}^{key} = \{ip_1, ip_2, \ldots, ip_J\}$ which is a list of

J number of current cellular users' IP addresses forming the a-Chord under the same station. Any cellular user entering the C-Chord system generates the *special key* by hashing its own base station Cell-ID and joins the corresponding a-Chord ring from the list $\mathcal{IP}^{key}$. $\mathcal{IP}^{gw} = \{ip_1, ip_2, \ldots, ip_{J'}\}$ denotes the list of IP addresses of the *successor* of the *special key* and adjacent *predecessors*. The cellular peer in the a-Chord ring maintains $\mathcal{IP}^{gw}$ that serves as the gate way towards the m-Chord. Cellular users' resources may be concealed in the a-Chord ring from the m-Chord nodes. To make cellular users a valuable trading partners with the peers in the m-Chord; some of those popular resources are hashed and published in the m-Chord through the peers in the list $\mathcal{IP}^{gw}$. Both m-Chord ring and a-Chord rings maintain each node's *successor* and hold node *successor*(k) accountable for key k in the dynamic network with peers joining and leaving all the time. For efficient lookup, the m-Chord ring and a-Chord rings also maintain *finger tables*. Number of peers in each a-Chord ring is far fewer than the number of Internet peers in the m-Chord. Each cellular peer, therefore, maintains a *finger table* with fewer entries.

4.2.1 Node's Joining Process

/* u_j from the j-th base station joins the network */
/* n' is an arbitrary node in the main Chord */
/* id^j_{cell} is the key id of j-th base station's Cell-ID */

u_j.**cu_join**(n')
if (n') **then**
  $n = n'.search\_special\_key(id^j_{cell})$;
  if $n \neq nil$ **then** /* u_j joins in j-th aChord */
    /* Copies the list of ip addresses */
    $\mathcal{IP}^{key}_j = get(id^j_{cell})$;
    /* Joins the j-th aChord */
    $u_j.join(ip)$ /* $ip \in \mathcal{IP}^{key}$ and currently active */
    Set $\mathcal{IP}^{gw}_j$;
    Update $\mathcal{IP}^{key}_j$ by replacing last entry with u_j's ip address;
  else /* u_j is the only node of j-th aChord */
    for $i = 1$ **to** m' **do**
      $finger[i].node = u_j$;
      $predecessor = u_j$;
    end
  end
else /* u_j is the only node in mChord */
  for $i = 1$ **to** m **do**
    $finger[i].node = u_j$;
    $predecessor = u_j$;
  end
end
/* ask node n to search for the key's id_k */
n.**search_special_key** (id_k)
$n' = find\_successor(id_k)$;
if n' *contains the key* **then return** n';
else return nil;

Fig. 4 Pseudo code for the cellular user joining process.

Peers may join (or leave) the system at any time in the dynamic network. Here we discuss how a cellular peer u_j from the j-th base station joins the P2P network. We denote id^j_{cell} as the m-Chord *id* of the j-th base station Cell-ID. When cellular node u_j wants to join the system, it is informed of any peer (cellular or Internet) by a bootstrap peer. In our design we do not impose any special property to the bootstrap peer, and therefore any existing peer may serve as the bootstrap peer. In case the bootstrap peer is a cellular peer, it can also inform the joining peer about any Internet peer n in the m-Chord. Then it contacts the Internet node n and searches the special key corresponding to its own Cell-ID. The Internet node n does not require any special properties to guide the cellular peer. C-Chord performs the following task when a cellular peer u_j joins the network.

- Find the *successor* of the special key id^j_{cell} on the m-Chord.
- Update and copy the data associated with the key id^j_{cell}. The data is simply the list of IP addresses of the cellular users participating in the a-Chord from the j-the base station.
- Join the a-Chord by contacting any of the cellular users from the list of IP addresses.

Fig. 4 depicts the pseudo code for the cellular user joining process. We now describe the node joining/leaving process in further details. Fig. 5 shows a partial view of a m-Chord along with the single a-Chord. Fig. 5(a) depicts that two cellular peers with *ids* 0, and 3 are connected with the m-Chord. Both cellular users bear the same base station Cell-ID (*special key id*) 2 which is stored in the m-Chord node 3. The data corresponding to *special key* 2 is only the list of IP addresses of these two cellular nodes already joined the a-Chord; i.e. $\mathcal{IP}^{key} = \{ip_0, ip_3, \ldots\}$. Key 1 is stored in cellular peer 3 accordingly. Fig. 5(b) shows the changes in the finger table and the special key when a new cellular peer with the same Cell-ID joins the a-Chord at *id* 2. For example, consider the joining cellular user is informed of the m-Chord by a m-Chord node. Since it bears the same Cell-ID, it finds the successor (m-Chord node 3) of the *special key* 2. Cellular user with *id* 2 then gets the data corresponding to *special key* 2 and updates its own IP address by modifying the data as $\mathcal{IP}^{key} = \{ip_2, ip_0, ip_3, \ldots\}$. It also initiates its *finger table*. Other peers then modify their *finger tables* accordingly. Key 1 is now stored in cellular peer 2. Fig. 5(c) depicts the leaving process of cellular peer 2.

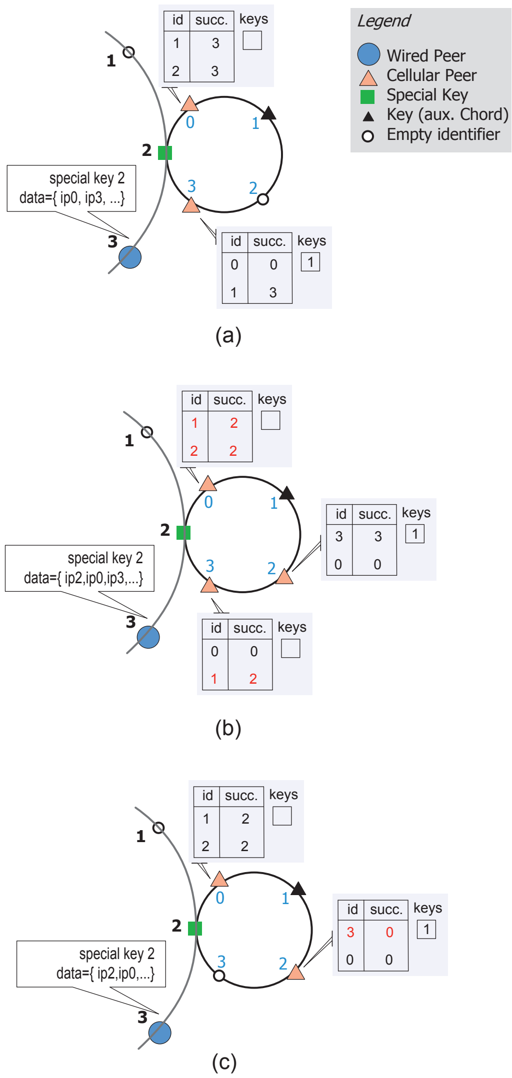

Fig. 5 An illustrative example of node joining/leaving process. (a) An *a*-Chord with two cellular peers with *ids* 0, and 3. (b) A new cellular peer joins the cellular network with *id* 2. (b) Cellular peer with *id* 3 leaves the system.

4.2.2 Lookup process of a cellular node

If the cellular node in the C-Chord system prefers to download from the Internet peer, it sends query to successor of the *id* corresponding to its own base station's Cell-ID or to any of the node from the list $\mathcal{IP}^{gw}$. Anyone might prompt to raise the scalability issue here and argue that search process might overwhelm the successor lists with queries. Bear in mind, as the cellular users increase so is the number of base stations and the number of *a*-Chord rings. Also, only some percentage of cellular users from each base station join the P2P system. Thus the users from the cellular networks are distributed and sorted by their Cell-ID. This innate property of the cellular network provides scalability to the C-Chord system. If the cellular user rather wants to avoid the Internet data penalty, it first tries to resolve a query within the *a*-Chord system and then forwards the query to the *m*-Chord.

4.2.3 Resilience to peer failures

In the C-Chord P2P system, the large *m*-Chord ring and all *a*-Chord rings inherit the property of resilience to peer failure from the base Chord system. Even though the *special key* may suffer from node failure; the detach *a*-Chord can rebuild the key due to the uniqueness of Cell-ID. In case new peers form another *a*-Chord by hashing the same base station Cell-ID, the old *a*-Chord can reclaim and reunite with the new *a*-Chord by the virtue of the uniqueness of the *special key*. This inborn property provides robustness to the C-Chord P2P system. Also, *special key* failure only affects the corresponding small *a*-Chord ring, not the large *m*-Chord ring and other *a*-Chord rings in the C-Chord P2P system.

4.3 Peer selection strategy

Here we present the peer selection strategy that allows requesting peers to choose peers from the potential senders. First we describe how we realize the multiple senders' discovery. Then we formulate the peer selection strategy as a stable-marriage problem. We discuss the solutions to the stable-marriage problem as well. Finally, we provide an alternate heuristic algorithm to solve the problem.

4.3.1 Enabling multiple senders discovery

In DHT based lookup protocols, two basic operations are: $put(k, data)$ operation which stores a key k and its associated $data$; and $get(k)$ operation that retrieves the data associated with k. We make use of these operators to empower the lookup process to find multiple potential senders. When k is the *id* of a resource segment in the P2P system; $data = \{ip_1, ip_2, \ldots, ip_R\}$ with ip_r is the r-th potential sender's IP address that has the same file segment. The data always keeps at most R number of latest IP addresses. With $i < j$, ip_i is the more recent potential sender's IP address than the ip_j. Whenever a sender releases a file segment that has a key k, it registers its IP address in the first place in corresponding $data$; if some other senders' IP addresses already exist, those will be shifted in an ordered manner.

4.3.2 Stable marriage based peer selection algorithm

```
/* R = {r_i : 1 ≤ i ≤ N} is a list of requesting peers      */
/* S = {s_i : 1 ≤ i ≤ N} is a list of potential senders     */
/* Pref_i^r is the i-th requesting peer's preference list   */
/* Pref_i^s is the i-th potential sender's preference list  */
/* M = {< r_i, s_j >}_N is a set of stable match            */
Input   : {Pref_i^r}_N, {Pref_i^s}_N
Output  : M
M ← φ;
while ∃ r_i an unmatched requesting peer do
    s_j ← most preferred sender in Pref_i^r;
    if s_j is unpaired then
        < r_i, s_j > becomes an element of M;
    else                               /* ∃ < r_k, s_j >∈ M */
        if s_j prefers r_i to r_k then
            < r_i, s_j > becomes an element of M;
            r_k is unpaired;
        else
            < r_k, s_j > is still an element of M;
        end
    end
end
return M
```

Fig. 6 Pseudo code for the stable marriage peer selection scheme.

The SM algorithm finds a stable matching between men and women through a sequence of proposals from men to women. Both men and women have preference lists. A stable assignment is called optimal when there is no man, woman pair of which both have stimulus to elope. This algorithm was first introduced by Gale and Shapley [7] later extended to many variations by Gusfield and Irving [8]. In our peer selection problem, a man is analogous to the requesting peer, and a woman is analogous to the potential sender. Like a man proposes to a woman, the requesting peer sends request to the potential senders. The reason behind choosing SM algorithm to solve the peer selection problem is as follows:

- The SM algorithm leads to stable solution by increasing satisfaction as large as possible rather than aiming at total satisfaction. This property makes the solution feasible.
- SM algorithm can sustain to peer churn rate, with peer joining/leaving the system all the time. When any peer joins the system requesting a file, it gets its preference list from the lookup protocol. When a receiver leaves the system, the corresponding sender can update its preference list without passing any message on the network. If sender leaves the system, the corresponding receiver sends request to the next preferred potential sender.
- Like men/women, requesting-peers and potential-senders also have their own selfish (often conflicting) goals to optimize.
- Cooperation between requesting peers and potential senders is the key ingredient of this algorithm. This is suitable and similar to our peer selection algorithm as a requesting peer does not require/have other requesting peers' information. Similar statement is applicable in case of potential senders. Thus the algorithm has inherent immune to selfish behavior of peers.

Fig. 6 presents the pseudo code for the SM peer selection scheme with equal N number of requesting peers and potential senders. The requesting peer gets the list of potential senders by enabling multiple senders discovery as described in section (4.3.1). The wired sender replies with a probable data rate; and the wireless sender replies with its energy state in addition to the data rate. Then the receiving peer selfishly produces its own preference list as the following way:

- Cellular receiver prefers a wired sender over a wireless sender. This is due to the fact that it saves wireless bandwidth avoiding two-hop wireless communication.
- Among wired senders, the higher the data rate the more it is preferred.
- Among cellular senders, the sender within the same a-Chord is preferred over a sender from the other a-Chord. This way cellular user may download from local peers.
- When a cellular receiver finds more than one cellular senders with close data rate, one with better energy-state is more preferred.

Preparing preference is simple and straightforward. Optimum performance is achieved from stable marriage algorithm by making use of the individual preference. When any requesting peer sends request to the potential sender, it also sends the number of potential peers it has discovered. The rules for preparing a preference list for the sending peer is similar to that of the receiving peer with the terms 'sender' and 'receiver' exchangeable. However, sender always prefers a receiver with single potential peer over the receiver that has multiple potential senders disregarding other rules. This algorithm is easily extendable to unequal sets with modified stopping condition [8]. There the number of match will be equal to the number of elements of the smaller set. The solution is receiver-optimal and sender-pessimal. Sender-optimal solution can also be obtained by altering the position. SM peer selection algorithm requires lot of messages (See section (4.3.4); therefore we propose an alternative heuristic algorithm.

```
/* IP_s is a list of the discovered potential senders      */
/* CQ is a list of IP_s's connection quality               */
/* Pref_r is the requesting peer's preference list         */
/* ip_s is the most preferred sender's IP address          */
Receiver Side  : -
while download session is not over do
    IP_s ← PeerLookUp(key);
    CQ ← AcquireConnectQuality(IP_s);
    Pref_r ← PrepPrefList(CQ);
    while Pref_r is not empty do
        ip_s ← PushMostPref(Pref_r);
        SendContrPack(ip_s);
        SwitchingRequire()←= false;
        while SwitchingRequire() == false do
            ReceivedDataSegment(ip_s);
        end
    end
end

/* Pref_s is the potential sender's preference list        */
/* ip_r is the most preferred req. peer's IP address       */
Sender Side  : -
while ShareResource()== true do
    if ReceiveContrPack() == true from a req. peer ip_j
    then
        Pref_s = UpdatePrefList(ip_j);
        ip_r ← PushMostPref(Pref_s);
    end
    if ip_r == ip_j then          // ip_j is now most preferred
        if CurrConnection == false then
            SendDataSegment(ip_r);
        else
            TerminateCurrConnection();
            SendDataSegment(ip_r);
        end
    end
end
```

Fig. 7 Pseudo code for the proposed heuristic peer selection scheme.

4.3.3 An alternate heuristic peer selection algorithm

Here we propose an alternative heuristic peer selection scheme that is near-optimal and requires less number of messages. We loosely follow SM algorithm to reduce the number of messages and mainly rely on sender and receiver collaboration. Note that we just require updating the connection whenever a new connection is requested or an old connection terminated. Fig. 7 shows the pseudo code of the alternate dynamic peer selection algorithm. As usual, the requesting peer discovers multiple potential senders by PeerLookUp(key) operation. Then it acquires connection quality of the potential senders and thereby prepares a preference list. Then the requesting peer connects to the more preferred sender. All other potential senders remain as standby peers. In case the preference list is empty, the requesting peer searches for potential senders through previously mentioned lookup process. On the other hand, at the sender side, its preference list is updated whenever it receives download request or any active connection terminated. The requesting peer sends the number of potential senders it can connect with the connection request. Any requesting peer, that has only one discovered peer in the list, is the most preferred one. As we do not want to starve anyone by feeding someone that has alternative resources. If two or more requesting peers have the same number of discovered potential senders; the potential sender prefers the one with the higher data rate achievable.

4.3.4 Computational burden

We will now address the message overhead and the computational complexity issues of the SM and heuristic peer selection scheme. The SM peer selection algorithm requires to exchange on the order of $\mathcal{O}(n^2-n+1)$ messages before finding an optimal solution; where n is the number of requesting peers or potential senders. Therefore, this method is unrealistic with large n. On the other hand, our proposed heuristic algorithm requires fewer messages and lesser computer complexity. This method requires exchanging only the order of $\mathcal{O}(n)$ messages; when a requesting peer already has discovered potential peers, and tries to choose one of them intelligently. The message overhead is lighter when compared to the size of file segments shared.

5 Performance Measurement

We first evaluate the performance of our proposed C-Chord P2P system in terms of path-length (number of hop-count) per lookup request. Then we numerically analyze the C-Chord based P2P system in terms of throughput experienced by the cellular receivers and investigate the effects of peer selection strategy. Finally, we evaluate the performance of the peer selection module.

5.1 Hop-count per lookup request

We implement the C-Chord protocol iteratively. Any peer, searching for a specific key, sends queries to a series of nodes and each time advances closer to the *successor*. Mean hop-count between two arbitrary nodes, in resolving a given query, influences the performance of the routing protocol and the scalability of the P2P system; as the number of participating peers could be large. The base Chord model in [22] has mean hop-count $\frac{1}{2}\log_2 N$ in practice, where N is the network size. When a cellular user requests a query from the a-Chord ring of size N_a, the mean hop-count to resolve the query is

$$\frac{1}{2}p\log_2 N_a + \frac{1}{2}(1-p)\log_2 N_m \quad (1)$$

where p is the probability that the successor exists in the a-Chord and N_m is the size of the m-Chord.

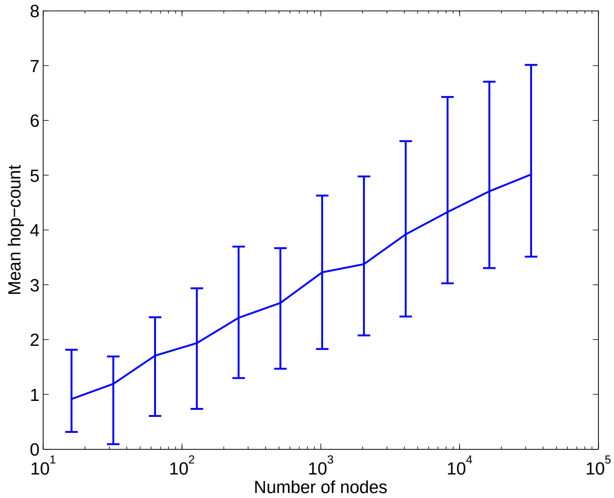

Fig. 8 Mean hop-count as a function of network size for $p = 0.5$.

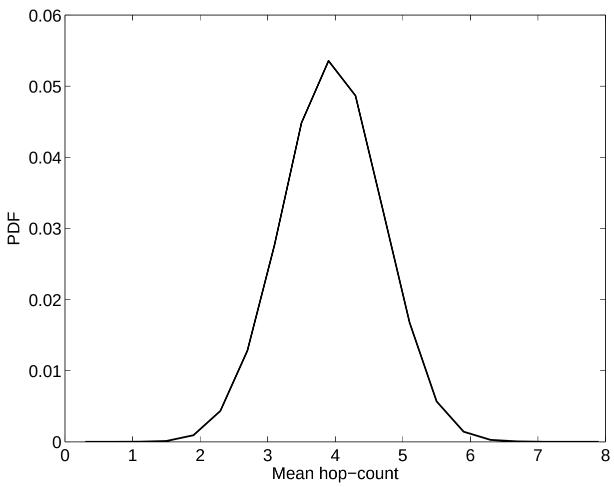

Fig. 9 The PDF of the mean hop-count with 2^{12} nodes. 50% of the nodes are cellular peers. The number of base stations or a-Chord rings is 32, each a-Chord consists of 64 peers.

In our simulation, the number of nodes in the overlay network is $N = 2^k$ that store 100×2^k keys. We arbitrary consider 50% of N nodes are cellular peers to highlight the effects of nodes from the cellular network. We vary k from 4 to 15, and also increase the number of base stations as the number of nodes from the cellular network increases. Each node, from the a-Chord, randomly requests a query for a set of keys, and we calculate the hop-count to resolve each query. First we illustrate the lookup efficiency of our C-Chord system; and then we compare the C-Chord model with that of the base Chord. Fig. 8 shows the mean hop-count as a function of k for $p = 0.5$. The error bar shows the 1st and 99th percentiles. As expected the mean hop-count increases with network size according to Eq. (1) approximately. For example with $N_m = 2^{11}$, $N_a = 32$, and $p = 0.5$; the mean hop-count is 4.0 approximately. Fig. 9 shows the probability density function of the hop-count for this network size.

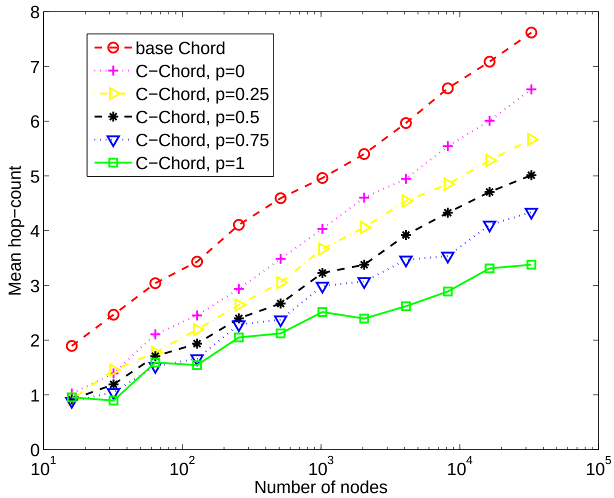

Fig. 10 Comparison of mean hop-count between base Chord and C-Chord with different values of p.

We also compare the mean hop-count per lookup request between the base Chord and the C-Chord P2P system. Since the cellular node, from the a-Chord ring, first sends a query to its existing a-Chord ring, and then to the larger m-Chord ring; the mean hop-count decreases as the probability of the *successor*'s presence in the a-Chord ring increases. Our proposed C-Chord system shows improve performances with smaller hop-count per lookup query over the base Chord system. Hence, the C-Chord P2P system integrates cellular network efficiently.

5.2 Network traffic follows in the P2P system

We also analyze C-Chord system to investigate the effects of selecting peers either from the same base station or from the outside of the Internet gateway. We implement the C-Chord system on several different networks that include peers from the Internet and also from the cellular networks. For each network, we select few nodes randomly that download files of approximately 100 MB from different sources. The number of bits in the *id* space is 160 and 40 for the m-Chord and the a-Chord, respectively. We perform each experiment

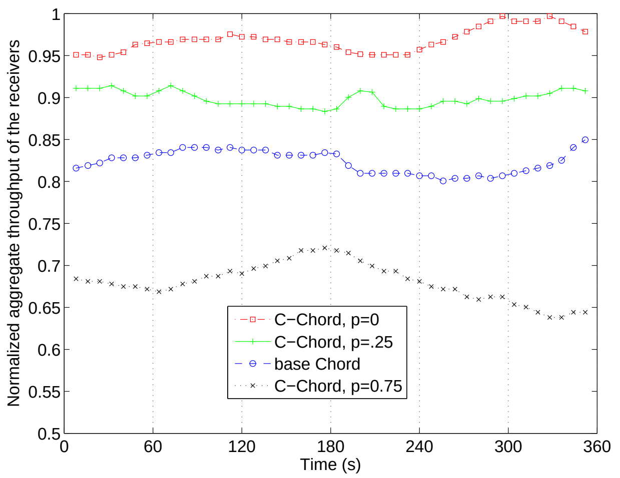

Fig. 11 The normalized aggregate throughput (per second) of the receivers for scenario '1'.

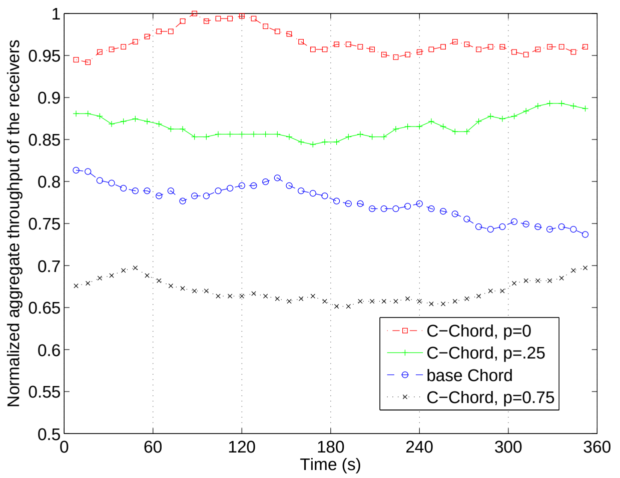

Fig. 12 The normalized aggregate throughput (per second) of the receivers for scenario '2'.

50 times tracing throughput of the receivers on each second. The aggregate throughput is averaged over all runs on each second. The physical links between Internet peers are generated by using a stochastic loss model [9]. The available bandwidth is set randomly in the range $[0.75R_0, 1.25R_0]$, where R_0 is the base bandwidth. We also set the maximum bandwidth of the cellular peers to $0.1R_0$, 10 times smaller than the wired network. We set $R_0 = 100$ $kbps$, which is realistic as sources are tapped while uploading. To capture the time varying capacity of the wireless channel, we follow the simulation settings as in [18]. As the simulation patterns remain similar, we present our results for two network scenarios. In both scenarios, the number of Internet peers and cellular peers are equal. In scenario '1', the number of Internet peers is 256, and there are 8 base stations each having 32 cellular users. While in scenario 2, the number of Internet peers is 512, and there are 16 base stations each having 32 cellular users. We pick randomly 5 pairs (receivers and senders) from scenario '1' and 10 pairs from scenario '2'. Fig. 11 and 12 show the normalized aggregate throughput (per second) of the receivers for scenario '1' and '2' respectively. If x senders are selected locally within the a-Chord ring from total y senders; the probability of finding the sources within the a-Chord ring, $p = x/y$. The receivers in all cases are chosen from cellular networks, as our primary interest is the performance of the cellular nodes. The run time of the simulation is 6 minutes excluding initial setup time.

With higher values in p (for example $p = 0.75$); the aggregate throughput is much lower as the receivers of the cellular networks download most of the contents locally. Some of the sources suffer from time varying wireless capacity and lower upload bandwidth. The base Chord P2P system also performs worse in terms of aggregate throughput than that of C-Chord P2P system with lower values of p. In the base Chord P2P system, the receivers cannot differentiate between cellular peers and Internet peers. The aggregate throughput is the highest when peer choose (or are forced) to download from the Internet peers only. That is when $p = 0$ in the C-Chord P2P system; all the receivers of the cellular networks download contents from the more stable Internet peers in m-Chord ring. However, in this case, the Internet data penalty is the highest.

5.3 Performance of peer selection algorithm

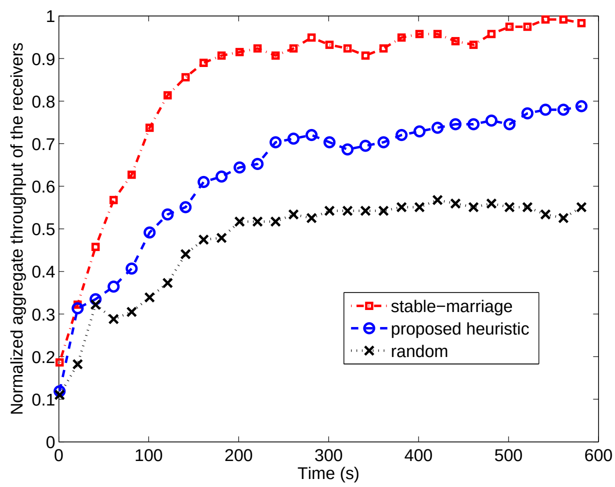

Fig. 13 The normalized aggregate throughput (per second) of the receivers for scenario '1'.

We consider 10 scenarios and pick 10 receivers randomly. Again as our primary interest is the performance

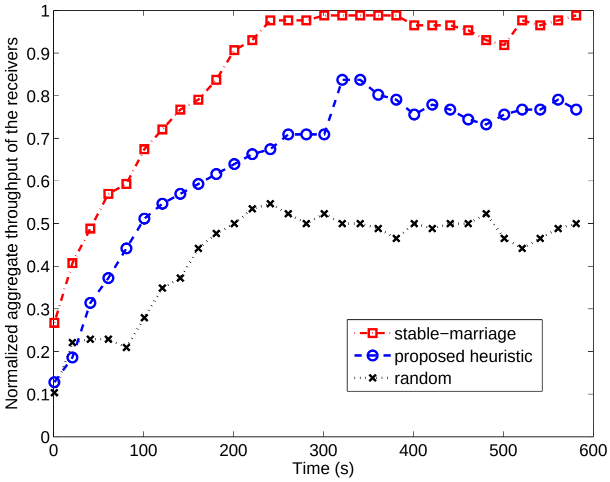

Fig. 14 The normalized aggregate throughput (per second) of the receivers for scenario '2'.

of the cellular nodes, the receivers in all cases are chosen from cellular networks. We deliberately put multiple potential peers at random locations. Not necessarily all requesting peers seek the same file segment, they may seek different file segments, and have few potential senders in common. In each case 10 requesting peers join the network at random time. We average the aggregate throughput experienced by the receivers. Since other eight scenarios show similar tendencies; we present simulation results only for two scenarios as shown in Fig. 13 and in Fig. 14. In both scenarios, average throughput, experienced by the receivers, increases as more requesting peers join the network. We also monitor the receivers' throughput after all 10 requesting peers join the network (after 300 seconds) to evaluate the stable state performance. Clearly, the heuristic peer selection algorithm performs better in terms received throughput than that of the random peer selection algorithm. Although SM peer selection scheme performs better than that of the heuristic peer selection scheme, SM peer selection requires off-line computation each time any requesting peer arrives the network. Any small variation in the aggregate received throughput is due to time-varying capacity of the wireless network.

6 Conclusion

In this paper, we have proposed C-Chord P2P system that integrates the cellular users into the popular Chord P2P system by enforcing a certain desirable structure to the overlay network. The C-Chord has provided the cellular users a choice of downloading contents either from the Internet peers at a faster rate or from other cellular users from the same base station avoiding Internet data penalty.

In the simulation, we have shown that mean hop-count per lookup request decreases with the increase of the probability of finding peers within the a-Chord ring. This is due to the fact the size (in terms of number of nodes) of the a-Chord ring is much smaller than that of the m-Chord ring. Moreover, the proposed C-Chord system has shown improved performance with lesser mean hop-count than that of the traditional Chord system for all values of probability of finding peers within the a-Chord ring. In the C-Chord P2P system, the maintenance of routing structure is more efficient with lesser entries in the a-Chord's *finger table*. We also have numerically analyzed the C-Chord based P2P system in terms of throughput experienced by the cellular receivers and investigated the effects of peer selection strategy. The base Chord P2P system has shown worse performance in terms of aggregate throughput than that of C-Chord P2P system with lower probability of finding the peers within the same a-Chord. The aggregate throughput was the highest when cellular peers downloaded contents from the more stable Internet peers in m-Chord ring paying highest Internet data penalty. We have shown through simulation results that the proposed heuristic peer selection scheme significantly improves the receiving data rate when compared to random peer selection.

Although in our simulation, we have scheduled few peers leaving the system and few others join the system at random time; we have not measured the degree of fault tolerance. Measuring the degree of fault tolerance even with concurrent join/failure is our ongoing task. We do not set any criteria of when and to what extent a cellular peer should offer resources to the m-Chord. Intuitively a cellular user shares resources in accordance to its upload bandwidth and energy-state. When any cellular peer leaves a base station, it also leaves the corresponding a-Chord and joins the appropriate a-Chord. Although in our simulations, we consider roaming and mobility of the peers; we do not compare and measure the performance gap with that of a utopia scenario. Both aforementioned concerns demand scrutiny in future. Our C-Chord system identifies users from the same base station. Cellular users in close proximity may communicate directly, by employing D2D connectivity offloading base station load [25]. We are over enthusiastic to enable this D2D option within the a-Chord ring in future.

In the proposed peer selection algorithm, only the requesting peer sends request for content download; potential sender does not initiate any connection (metaphor to woman does not propose). That is why the proposed algorithm always produces a requesting peer-optimal, and a potential sender-pessimal pairing. With peer leaving the P2P system, the sender's preference list changes. It would be interesting to examine the performance of the algorithm where the potential sender can request for connection (analogous to woman can propose too). There already exists low complexity distributed stable-marriage (DSM) algorithm [13]. In future we like to investigate whether DSM algorithm could be applied to efficient peer selection scheme specially in the context of live video streaming.

References

1. Gnutella protocol development (2010,). URL http://rfc-gnutella.sourceforge.net/index.html/
2. Kazaa (2010,). URL http://www.kazaa.com/
3. Berl, A., de Meer, H.: Integration of mobile devices into popular peer-to-peer networks. In: Next Generation Internet Networks, pp. 1–9 (2009)
4. Canali, C., Renda, M.E., Santi, P., Burresi, S.: Enabling efficient peer-to-peer resource sharing in wireless mesh networks. IEEE Trans. Mobile Comput. **9**(3), 333–347 (2010)
5. Cisco: Visual networking index: Forecast and methodology, 2008-2013. White Paper (2009)
6. ElRakabawy, S.M., Lindemann, C.: Peer-to-peer file transfer in wireless mesh networks. In: Proc. Wireless on Demand Network System and Services (WONS), pp. 114–121 (2007)
7. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1), 9–15 (1962)
8. Gusfield, D., Irving, R.W.: The stable marriage problem : structure and algorithms. MIT Press,, Cambridge, Mass. : (1989)
9. Hefeeda, M., Habib, A., Botev, B., Xu, D., Bhargava, B.: PROMISE: peer-to-peer media streaming using collectcast. In: Proc. ACM International Conference on Multimedia, pp. 45–54. ACM (2003)
10. Hsieh, H.Y., Sivakumar, R.: On using peer-to-peer communication in cellular wireless data networks. IEEE Trans. Mobile Comput. **3**(1), 57–72 (2004)
11. Huang, C.M., Hsu, T.H., Hsu, M.F.: Network-aware P2P file sharing over the wireless mobile networks. IEEE J. Sel. Areas Commun. **25**(1), 204–210 (2007)
12. Kim, K., Park, D.: Mobile NodeID based P2P algorithm for the heterogeneous network. In: Second International Conference on Embedded Software and Systems, pp. 1–8 (2005)
13. Kipnis, A., Patt-Shamir, B.: A note on distributed stable matching. In: IEEE International Conference on Distributed Computing Systems (ICDCS) , p. 466 (2009)
14. Leung, A.K.H., Kwok, Y.K.: On localized application-driven topology control for energy-efficient wireless peer-to-peer file sharing. IEEE Trans. Mobile Comput. **7**(1), 66–80 (2008)
15. Li, J., Huang, L., Jia, W., Xiao, M., Du, P.: An efficient implementation of file sharing systems on the basis of WiMAX and Wi-Fi. In: Proc. IEEE International Conference on Mobile Adhoc and Sensor Systems, pp. 819–824 (2006)
16. Li, X., Ji, H., Zheng, R., Li, Y., Yu, F.R.: A novel team-centric peer selection scheme for distributed wireless P2P networks. In: IEEE WCNC 2009, pp. 1–5 (2009)
17. Mäenpää, J., Bolonio, J.J.: Performance of resource location; discovery (reload) on mobile phones. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2010)
18. Mogensen, P., Na, W., Kovacs, I.Z., Frederiksen, F., Pokhariyal, A., Pedersen, K.I., Kolding, T., Hugl, K., Kuusela, M.: LTE capacity compared to the shannon bound. In: IEEE Vehicular Technology Conference (VTC), Spring, pp. 1234–1238 (2007)
19. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), pp. 161–172. ACM (2001)
20. Rowstron, A.I.T., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale Peer-to-Peer systems. In: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, pp. 329–350 (2001)
21. Si, P., Yu, F.R., Ji, H., Leung, V.C.M.: Distributed multisource transmission in wireless mobile peer-to-peer networks: A restless-bandit approach. IEEE Trans. Veh. Technol. **59**(1), 420–430 (2010)
22. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), pp. 149–160 (2001)
23. Wang, S., Ji, H.: Realization of topology awareness in peer-to-peer wireless network. In: International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4 (2009)
24. Zou, D., Gan, Y.: The research of a structural p2p model based on mobile cellular networks. In: International Conference on Networking and Digital Society (ICNDS), pp. 64–67 (2009)
25. Zulhasnine, M., Huang, C., Srinivasan, A.: Efficient resource allocation for device-to-device communication underlaying LTE network. In: (accepted for publication) in IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Canada (2010)

Mohammad Zulhasnine received his B.Sc. (2002) in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology, Bangladesh and M.A.Sc. (2008) in Electrical and Computer Engineering from University of Waterloo, Canada. He is currently pursuing his Ph.D. degree in Systems and Computer Engineering at Carleton University, Canada. Prior to joining at Carleton University, he worked as a Lecturer at Jahangirnagar University, Bangladesh from 2002 to 2005. His research interests include peer-to-peer applications over wired and wireless networks, device-to-device communication and radio resource management.

Changcheng Huang received his B.Eng. in 1985, and M.Eng. in 1988, both in Electronic Engineering from Tsinghua University, Beijing, China. He received a Ph.D. degree in Electrical Engineering from Carleton University, Ottawa, Canada in 1997. From 1996 to 1998, he worked for Nortel Networks, Ottawa, Canada where he was a systems engineering specialist. He was a systems engineer, and network architect in the Optical Networking Group of Tellabs, Illinois, USA during the period of 1998 to 2000. Since July 2000, he has been with the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada where he is currently an associate professor. Dr. Huang won the Canada Foundation for Innovation (CFI) new opportunity award for building an optical network laboratory in 2001. He was an associate editor of IEEE COMMUNICATIONS LETTERS from 2004 to 2006. He is currently a senior member of IEEE.

Dr. Anand Srinivasan is an expert in system and network engineering for IP based wireless networks. With over 20 years of experience in system design and network planning for large scale wired and wireless networks, Dr. Srinivasan is the principal architect developing EION's innovative mobile Ad Hoc networking technology and award winning WiMAX and LTE product line. In addition to his work with EION, Dr. Srinivasan is an Adjunct Research Professor in the Department of System and Computer Engineering at Carleton University. In this role he has been responsible for the stewardship of various projects with the Canadian Space Agency, Precarn, the Ontario Centers of Excellence and NSERC. Prior to joining EION, Dr. Srinivasan worked as a technical lead in the wireless and optical divisions at Nortel Networks. His efforts were instrumental in optimizing the performance of many of their flagship products, including the popular One Meg Modem. Dr. Srinivasan holds a Ph.D and M.Sc. in computer science from the University of Victoria, British Columbia. He has published over fifty papers in the areas of operating systems, distributed systems, fault-tolerance and optimization and holds two patents. He is a member of IEEE and AIAA and serves regularly in several research panels and external conferences including the ITC and Canadian UAV Forum.