

# United we find: Enabling Mobile Devices to Cooperate for Efficient Neighbor Discovery

Mehedi Bakht, John Carlson, Alexander Loeb, Robin Kravets  
Department of Computer Science, University of Illinois at Urbana-Champaign  
{mbakht2,carlson5,aloeb2,rhk}@illinois.edu

## ABSTRACT

The recent surge in the use of mobile devices have opened up new avenues for communication. While most existing applications designed to exploit this potential are infrastructure based, there is a growing trend to leverage physical proximity between end-users to enable direct peer-to-peer communication. However, the success of these applications relies on the ability to efficiently detect contact opportunities. Devices that participate in such opportunistic communication often come equipped with multiple radios. For an individual node, performing neighbor discovery can be too expensive with a high-power, long-range radio (e.g., Wi-Fi). On the other hand, relying only on a low-power, short-range radio for detecting neighbors results in significantly fewer available contacts. To mitigate this problem, we have developed CQuest, a novel scheme for more efficient long-range neighbor discovery that leverages the clustering of nodes as well as the radio heterogeneity of mobile devices. The basic idea is that coordination over a low-power, short-range radio can help clustered nodes distribute the load of high-power, long-range scanning. We present results from extensive simulation that shows CQuest discovers significantly more contacts than a low-power only scheme but without incurring the high energy cost usually associated with long-range discovery. We also present results and experience from a successful implementation of the protocol on a testbed of Android G1/G2 phones that shows the feasibility of the protocol in a real network.

## 1. INTRODUCTION

The recent explosion in person mobile communication devices (e.g., smartphones, PDAs, tablets, ebook readers, etc.) has not only enabled users to remain connected nearly all of the time but also presents new mobile peer-to-peer communication opportunities. Although infrastructure-based communication satisfies many of the user's needs, it is not sufficient in some scenarios and over kill in others. On one side, some devices only have Wi-Fi. In the absence of Wi-Fi

access points, these devices become incapable of any communication even if they are in close proximity to other devices. On the other side, if peer-to-peer communication can satisfy the user's communication requirements, it can eliminate the need to use limited and/or expensive data plans. For example, if such an opportunistic network based on mobile peers can establish a minimum level of connectivity and communication, it can support applications like proximity-based social networking [6, 16].

While it is easy to imagine two people's phones exchanging local weather or cooperating in a local game, the first step to establishing such device-to-device communication is discovery of contact opportunities to enable these exchanges. Given the unpredictable and unplanned mobility of nodes in real-life networks, discovery needs to be a continuous process. Unfortunately, searching for neighbors using a high-power (HP) radio like Wi-Fi is extremely expensive energy-wise, as previous experiments have shown (e.g., Mobiclique [11]). As a result of this expense, most distributed mobile social networking applications limit their discovery to the use of a low-power (LP) radio like Bluetooth (e.g., Mobiclique [11], PeopleNet [9]), significantly reducing energy costs. However, this reduction comes at the cost of a significant reduction in the discovery of neighbors, and so contact opportunities, due to the lower range of LP radios. The ultimate result of dismissing the use of the HP radio is degraded application performance and unsatisfied users.

Where current approaches dismiss the use of the HP radio, we present CQuest, a neighbor discovery protocol that embraces the use of the HP radio during neighbor discovery without incurring the high overhead. To reduce energy costs, the design of CQuest leverages the natural phenomenon of clustering in social networks, in which nodes often gather around different locations like classrooms, coffee-shops, bus stops, etc. While a LP short-range radio is sufficient for the nodes inside such a cluster to discover each other and form small communication groups [15], nearby neighbors only reachable via the long-range HP radio remain undiscovered. To reach these neighbors, CQuest enables cooperative scanning over the set of HP radios in a cluster through coordination over the LP radio.

CQuest is designed around a simple distributed approach that enables nodes to cooperate using only local knowledge about their neighborhood and incurring minimal overhead. Our simulation results show that such cooperation results in a dramatic reduction in energy consumption, up to 50% over an uncoordinated HP-based approach, without significantly giving up on the HP radio contacts. Additionally,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HotMobile'12 February 28–29, 2012, San Diego, CA, USA.  
Copyright 2012 ACM 978-1-4503-1207-3 ...\$10.00.

we show that this minimal reduction in contacts has little or no impact on opportunistic routing and that CQuest ultimately reduces the average communication latency. Obviously, the integration of the HP radio results in higher energy consumption than an LP-only solution. However, we show that the improvement in communication is dramatic, over 100% improvement in delivery ratio in our test scenarios. To evaluate the feasibility of CQuest and gage the actual amount of energy savings possible, we implemented CQuest on a testbed of Android phones. Our evaluation shows that CQuest can indeed enable long-range neighbor discovery on commercial off-the-shelf hardware at a significantly reduced energy cost.

The rest of this paper is as follows. Section 2 discusses radio heterogeneity and clustering in wireless networks. In Section 3, we present a detailed description of CQuest. Section 4 presents our simulation-based performance evaluation of CQuest in comparison to other cluster-agnostic schemes. A prototype implementation of CQuest on a smartphone testbed is described in Section 5. We conclude and outline the direction of our future research in Section 6.

## 2. EXPLOITING RADIO HETEROGENEITY

Most mobile communication devices now come equipped with multiple wireless interfaces (e.g., Bluetooth, Wi-Fi) that significantly vary in terms of energy-efficiency and transmission range. While both radios can be used for discovering and communicating with nearby nodes, current approaches only use the LP radio at the cost of significantly restricting the reach of a node. Instead of forcing a choice between between the radios and trading energy for range or the other way round, a middle ground can be found by taking a approach that coordinates the use of both radios, enabling communication and saving energy [4, 13, 3]. However, all of these approaches restrict discovery to the range of the LP radio, reducing the effective performance of mobile social network applications.

A naive approach to discovering distant neighbors would be to let each node in the network independently use their HP radio for discovery. However, this approach is inefficient because of a combination of two factors - clustering and radio heterogeneity. The mobility of nodes in real life often leads to the formation of temporary clusters [15, 7] where members of a cluster can reach each other through a LP radio. Because of the range disparity between HP and LP radios, the nodes within an LP neighborhood are very likely to discover the same set of HP neighbors.

To better understand the extent of overlap, let us consider a simple scenario consisting of two LP neighbors,  $A$  and  $B$ . For simplicity, assume the LP and HP transmission ranges are unit disks with radius  $r$  and  $R$  respectively. Assuming that  $A$  and  $B$  are separated by the maximum distance  $r$ , the intersection of their HP radio ranges is given by the following expression:

$$A_{intersection} = 2 \cdot R^2 \cdot \text{acos}\left(\frac{r}{2 \cdot R}\right) - \frac{1}{2} \cdot r \cdot \sqrt{4 \cdot R^2 - r^2}.$$

For typical transmission ranges of Wi-Fi and Bluetooth,  $R = 100$  m and  $r = 10$  m respectively, the ratio of the intersection area to the HP radio's transmission range is .936. Essentially, this means that for any node  $C$  which is an HP neighbor of  $A$ , there is very high probability (more

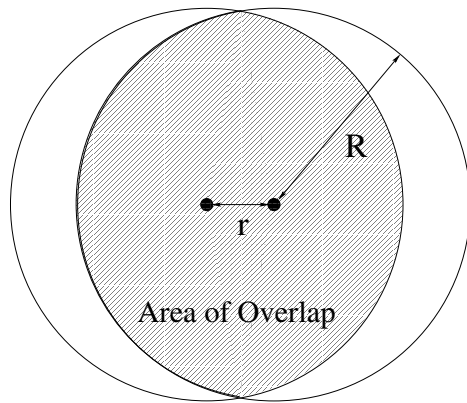


Figure 1: Area of Common HP Neighbors

than 93%) that it will be an HP neighbor of  $B$  as well (see Figure 2).

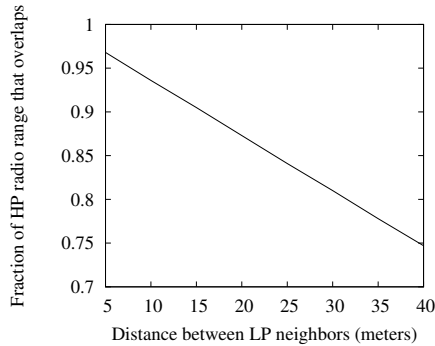
Given this redundancy, not all nodes in the network need to discover HP neighbors at the same time. Instead, HP discovery can be performed by a *subset* of nodes, which then disseminate the resulting information to non-discovery nodes. Since discovery is performed by scanning the wireless channel, we call this the *scanner set*,  $S_t$ , at time  $t$ . To enable high-probability discovery of HP neighbors, the nodes in the scanner set should provide coverage of the union of the HP neighborhoods of the cooperating nodes. Since solving such a coverage problem is NP-Hard even in the best of conditions, we consider probabilistic coverage, where two nodes cover each other with high probability as long as they are direct LP neighbors.

Given such probabilistic coverage,  $S_t$  can provide coverage of the given area as long as any node in the network is either a scanner or has at least one LP neighbor that is a scanner. Essentially,  $S_t$  is a dominating set. To achieve maximum energy savings,  $S_t$  should be a minimal dominating set at any given time. Additionally, to load balance energy consumption,  $S_t$  should change over time so that the load gets distributed as evenly as possible.

However, finding and maintaining a dominating set in a dynamic network with unpredictable mobility is very challenging and requires central coordination. Ensuring the minimality of the dominating set and achieving fairness makes it all the more difficult. Additionally, the mobility in the network can frequently compensate for inefficiencies in the dominating set, eliminating the need for exact solutions. In response, we next present CQuest, a collaborative protocol that effectively leverages radio heterogeneity for performing *neighbor discovery* over the high-power radio at significantly reduced energy costs, which ultimately leads to more effective opportunistic communication.

## 3. CQUEST

The basic goal of CQuest is to enable efficient neighbor discovery by leveraging the diverse characteristics of multiple radios. CQuest has three major components: scanner set selection, which enables cooperative discovery of HP neighbors at reduced energy costs, HP and LP scanning, which provides the actual discovery of neighbors, and neighbor maintenance, which enables nodes to exchange the results of HP scanning over the LP radio so that non-scanner



**Figure 2: Probability of a common HP neighbor**

nodes can learn about their potential HP neighbors without scanning themselves.

### 3.1 Scanner Set Selection

The main challenge for scanner set selection is providing a light-weight distributed solution to probabilistic coverage in a dynamic network. In a static network,  $S_t$  can be determined during network set up. However, in a network with mobile nodes,  $S_t$  needs to be updated periodically to maintain coverage. Additionally, composition of  $S_t$  needs to change periodically to distribute the load of scanning. Given these requirements, CQuest assumes a discrete-time model, where time is divided into rounds of equal length and  $S_t$  is determined for each round. The exact definition of a round in time units can be determined based on actual network characteristics and is not tied to the functioning of the protocol.

The goal of scanner selection in CQuest is to construct  $S_t$  for each round, where  $S_t$  is dominating for that round. Due to the dynamic nature of mobile networks and need to reduce control overhead, CQuest takes a purely local approach that enables each node to independently determine whether or not it should be a member of  $S_t$  based on simple contention-based approach. The result is an approximation of a dominating set that errs on the side of a small amount of redundancy in  $S_t$ .

In CQuest, each node determines its own membership in  $S_t$  by looking at its LP neighbors. Only if none of its LP neighbors is in  $S_t$ , the node includes itself in the scanner set. This type of approach is analogous to MAC-level contention resolution in multi-hop wireless networks. In MAC protocols, nodes use carrier sensing (physical, and sometimes virtual) to see if any other node in the vicinity is transmitting. If not, it proceeds to transmit. Otherwise, it waits for the transmission to end. Similarly, the basic idea for scanner selection in CQuest is that a node checks to see if any of its LP neighbors is going to do a scan for HP neighbors in the next round (i.e., is a member of  $S_t$ ). If yes, the node does not scan in the next round to reduce redundancy. Otherwise, the node starts scanning in the next round.

To reduce the possibility of collisions or redundant scanning, CQuest uses a contention phase at the end of each scanning round. The contention phase is divided into contention slots, where the number of slots is a configurable protocol parameter called *windowSize*. At the beginning of each contention phase, nodes randomly pick a slot between 0 and *windowSize* and broadcast a contention packet in that slot over the LP radio. If a node receives a contention

packet before its own selected slot, the node decides that it has “lost” that contention phase and one of its neighbors is in  $S_t$ . Otherwise, if a node does not get any contention packet before its own selected slot comes up, it broadcasts a contention packet itself and considers itself as the winner of that contention phase and includes itself in  $S_t$ .

Obviously, using this type of contention resolution may result in occasional collisions and multiple LP neighbors could end up scanning in the same round. To reduce control overhead and to maintain the completely distributed nature of the protocol, CQuest does not require a minimal dominating set and allows “redundant” scanning. Similarly, CQuest trades off simplicity and reduced control overhead for the guarantee of coverage in the face of mobility. If a scanner node leaves a cluster, there may be a period of time without complete discovery coverage.

The final part of scanner selection is load balancing. If the same set of nodes always scans, their energy will drain quickly. Instead, the cost of HP discovery needs to be shared across the LP cluster. To achieve fairness, CQuest attempts to improve the chances of a node being a scanner in a round if it was not selected in the previous round(s). While different techniques can be applied to achieve this objective, CQuest uses an exponentially decreasing contention window size. Initially, nodes start with a *windowSize* equal to *max-window-size*. Every time a node fails to win a contention phase, it decreases the *windowSize* by reducing it by half, until it reaches the *min-window-size*. On the other hand, if the node wins a contention phase, it resets the *windowSize* to *max-window-size*.

### 3.2 HP and LP Scanning

After selection of the HP scanner set through coordination over the LP radio, the next step is to scan for neighbors using both of the radios.

In each round, scanners use their HP radios to discover any node in their HP neighborhood and non-scanners turn off their HP radios. The actual protocol used for scanning and discovery is determined by the specific network environment. In a synchronous network, the discovery protocol can be as simple as using a beacon with a fixed period, and so a round could be defined as some pre-defined number of such periods. For asynchronous networks, existing asynchronous neighbor discovery protocols (i.e., U-connect [8]) can be used.

The LP radio can run discovery process of its own anytime except for during the contention phase. Again, the exact method for discovery depends on the radio used (e.g., Zigbee, Bluetooth), application requirements, etc.

### 3.3 Maintenance and Exchange of Neighbor Information

CQuest supports direct neighbor discovery (either LP or HP) through beaconing, or indirectly through disseminated neighbor information from other nodes. HP discovery beacons include the IDs of all LP neighbors. Similarly, LP discovery beacons include the IDs of all HP neighbors. Once a node gets added to the neighbor database, subsequent discoveries refresh that information. CQuest uses two thresholds to determine the staleness of an entry based on the last refresh time. If the time elapsed since the last refresh time is more than *freshness-threshold*, the entry still remains in the database but is not included as part of neighbor information

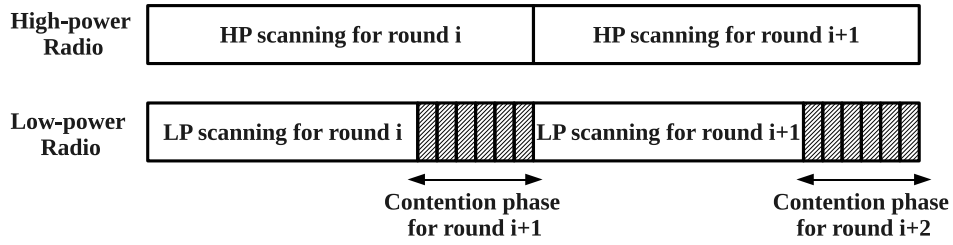


Figure 3: Contention Phase

exchanged. On the other hand, if the time elapsed since the entry was last refreshed exceeds  $expiration\_threshold$ , the entry gets completely removed from the database. Obviously,  $expiration\_threshold \geq freshness\_threshold$ .

## 4. EVALUATION

The goal of our evaluation is to two-fold. First, we evaluate the effectiveness of CQuest in terms of energy consumption and successful neighbor discovery. Second, we evaluate the impact of the slightly reduced contact opportunities of CQuest as compared to the uncoordinated HP approach by using a common DTN routing protocol.

To evaluate CQuest, we consider three metrics. First, the **Number of Successful Discoveries** captures the effectiveness of a given discovery protocol. Second, **False Positives** captures the effect of indirect discovery, which can lead to discovery of contact opportunities that do not exist. False positives can happen for two reasons: stale information, where a neighbor may have simply moved away, or lack of overlap, which is caused by CQuest’s probabilistic assumptions about coverage. Finally, the **Total Energy** consumed during the use of each protocol determines its ultimate efficiency. For the HP interface, we include the energy consumed for on-to-off transitions, beacon transmissions, beacon receptions, idle time spent during an active slot, and also off-to-on transitions. For the LP interface, we include the energy consumption of beacon transmissions and receptions. We omit the idle energy consumption for the LP radio, since the LP interface is on all of the time for all protocols.

To see the effect of social mobility and the formation of clusters, we use the community-based mobility model described in [10], which captures user community structure in an implicit way by quantifying the relationship between the nodes of the network. Essentially, each node is associated with some set of nodes that belong to its community. The weight of the edge connecting two nodes signifies the strength of their social relationship. The simulation of this model generates traces with characteristics similar to that of real life traces obtained from Intel Research in Cambridge. We vary the number of nodes from 40 to 100 to study how CQuest performs in networks with different densities.

We compare the following schemes with CQuest:

**LP:** Nodes use only their LP radio to detect contacts. The LP interface is always on and periodic beacons are broadcast to enable discovery. This is the most common single radio approach used by most current mobile social networking applications (e.g., [11, 12]). This approach uses the least amount of energy and provides a baseline for the minimum number of contacts discoverable.

**HP:** Nodes use asynchronous neighbor discovery over their HP radios and keep their LP radios completely off. For our evaluation, discovery was performed using U-Connect with  $p = 19$ .

**LP-HP:** Nodes use both radios for neighbor discovery in a combination of LP and HP, but without any coordination-ordination.

All protocols were implemented on the ns2 [1] simulator. To simulate dual-radio nodes, we used the NS-MIRACLE extension [5]. However, NS-MIRACLE does not support Bluetooth. Hence, for the short-range radio, we use a radio that has energy, transmission range, and bandwidth characteristics similar to Bluetooth but is capable of broadcasting. We discuss how CQuest can be adapted to work with a regular Bluetooth radio in Section 5. For all graphs, error bars denote 95% confidence intervals.

**Successful Detection of Contacts:** For all node densities, CQuest discovers significantly more contacts than the LP scheme, with a minimum improvement of around 250% (see Figure 4(a)). On the other hand, in comparison to HP-LP and HP, even in the worst case, CQuest discovers 15%-20% fewer contacts, since it does not guarantee coverage.

**Energy:** We can see from Figure 4(b) that CQuest achieves an energy savings of around 23% when node density is lowest (discovery success rate of CQuest was only around 1%-5% lower for the same density). As expected, the savings increases with an increase in node density, and for 100 nodes, the energy savings exceeds 50%.

**False Positives:** We calculated the number of such false positives as a percentage of successful discoveries. For all densities, the fraction of false positives remained very small and almost constant in the interval 2%-2.5% (see Figure 4(c)). This confirms that two LP neighboring nodes often share the same HP neighborhood. We used the default speed from the community-based model since that was shown to mimic real world characteristics well. As future work, we would like to evaluate how changing the speed affects this value.

**Effectiveness of Discovered Contacts:** The primary motivation for enabling the discovery of more neighbors is to enhance the performance of protocols that depend on opportunistic contacts. To evaluate the extent to which CQuest achieves this goal, the contact traces from the simulation of the different discovery schemes were given as an input to an opportunistic routing protocol. To compare the impact of different discovery schemes, we looked at *Delivery ratio*, the percentage of total generated messages that eventually get delivered.

For our simulations, we used the Opportunistic Network Environment (ONE) simulator [2] and chose Spray and Wait [14] as the routing protocol because of its resource friendliness.

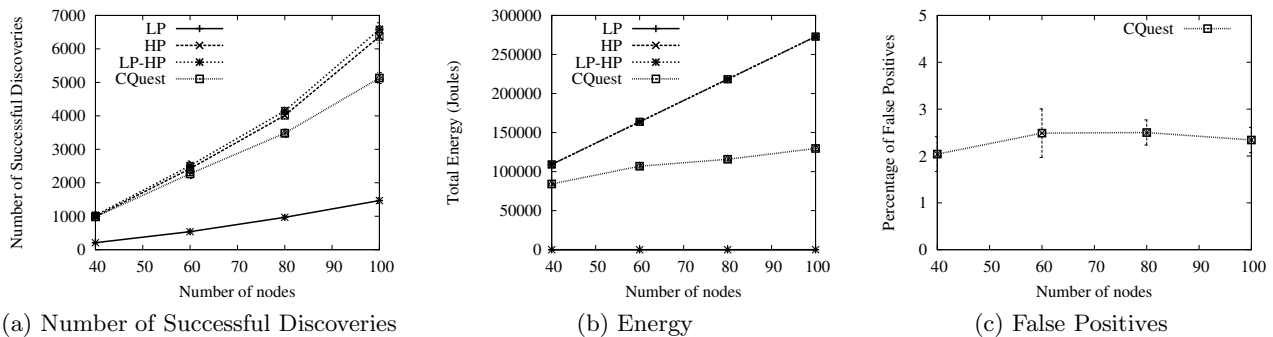


Figure 4: Comparison of different discovery schemes

One message was generated at a randomly selected node every second and the destination for that message was also chosen randomly. For all runs, the message size was varied from 1 KB to 10 KB. Each data point is the average of 10 runs. Since we are interested only in seeing the impact of contacts, the buffer size was set to a very large value (500 MB) so that performance of the protocols do not get affected by buffer constraints.

Simulation Area	500m × 500m
Range of the HP radio	100 m
Range of the LP radio	10 m
beacon period of the LP radio	5 seconds
Protocol used for Wi-Fi Discovery	U-Connect (p=19)

Table 1: Simulation Settings

Irrespective of number of nodes in the scenario, schemes that use the HP radio (i.e., HP, LP-HP, CQuest) deliver many more messages than LP (more than a 100% improvement in delivery ratio for all densities) (see Figure 5). However, more important to evaluate is the performance of CQuest in comparison to HP and HP-LP, since, as we have seen earlier CQuest fails to detect some contact opportunities since, unlike those schemes, all nodes are not scanning all of the time. The delivery ratio indicates the significance of those missed contacts. As we can see, although CQuest discovered fewer contacts than HP and LP-HP, its delivery probability is almost as good as the two protocols. This means that the contacts missed by CQuest were probably too short to be taken advantage of anyway.

## 5. IMPLEMENTATION

To evaluate performance in a real setting, CQuest was implemented on a testbed of 15 Google G1 and G2 Developer phones running a modified version of CyanogenMod 6.0<sup>1</sup> which supports Android 2.2. The biggest challenge of the implementation was adapting CQuest to the non-broadcast medium of Bluetooth.

The original design of CQuest assumed a broadcast capable LP radio to support the contention phase. To work with Bluetooth, the contention phase was modified from implicit coordination via broadcast to explicit coordination via directly exchanged messages. Specifically, with a broadcast medium, nodes receive messages and act on the information they contain without any further communication required. However, Bluetooth requires that a connection be

established between two nodes to exchange messages, making the communication explicit. To solve this problem in the Bluetooth implementation using unicast connections, CQuest was modified so that the scanner for any round  $i$ , denoted  $scanner_i$ , becomes responsible for running the contention phase for round  $i+1$ . First, at the beginning of round  $i$ ,  $scanner_i$  performs Bluetooth device discovery to find all nodes in its Bluetooth neighborhood that run the CQuest protocol. Then, at the beginning of the contention phase, each node selects a slot value for itself as described before. However, no contention packet gets sent based on the chosen value. Instead,  $scanner_i$ , based on the list of Bluetooth neighbors it has acquired earlier in the round, connects directly to each member of that list one by one and asks each neighbor what value it has chosen. Then, after querying all neighbors,  $scanner_i$  determines who the scanner will be for round  $i+1$  (in case of a tie, it chooses randomly) and contacts that node directly to inform it that it should perform scanning for the next round. If the Bluetooth cluster is multi-hop, it becomes all the more difficult to execute the contention phase successfully.

CQuest had to be further modified to deal with the scanner leaving its cluster because, unlike the original protocol, a contention phase over Bluetooth cannot take place in the absence of a scanner. To handle this issue, each node keeps a timer based on the expected interval between contention phases, and the time by which it can expect to be contacted if there was a scanner nearby. When that timer expires, nodes probabilistically decide to “run” the contention phase themselves after a randomly chosen period, unless any other scanner contacts it before then.

Since our main goal is energy-efficiency, we evaluated whether CQuest can enable the phones to coordinate and use less energy. For this experiment, we used a static cluster and varied the size of the cluster size from 1 to 7. The cluster was a clique (i.e., all phones were in Bluetooth range of each other). If the nodes coordinate successfully, for an  $n$ -node clique, each node should spend only  $\frac{1}{n}$  of the energy it spends when it scans by itself. For measuring energy savings, we looked at how long it took on average for the battery levels of the phones to decrease from 75% to 50%. The protocol used for Wi-Fi discovery was U-Connect with  $p = 7$ . In our experiments, the phones were able to successfully coordinate and the average time to go from 75% to 50% grew almost linearly with an increase in cluster size (see Figure 6). Also, as a base case comparison, we show the time it takes for

<sup>1</sup>See <http://www.cyanogenmod.com/>

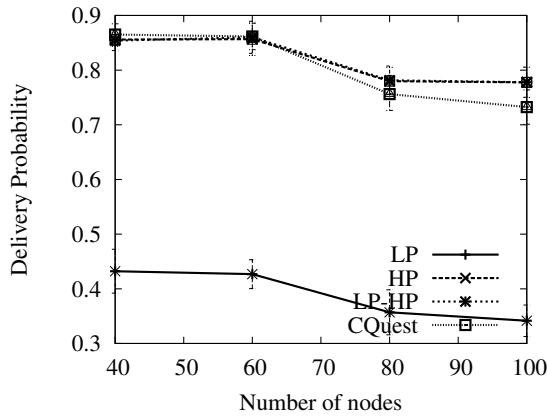


Figure 5: Delivery Ratio

a similar change in battery level when the phones perform Wi-Fi discovery without any cooperation.

## 6. CONCLUSION & FUTURE WORK

In this paper, we present CQquest, a new coordinated neighbor discovery protocol that addresses the problem of neighbor discovery in opportunistic networks by leveraging the clustering of nodes and the presence of radio heterogeneity. Extensive simulation results show that CQquest discovers almost the same number of contacts as a regular Wi-Fi discovery protocol, while reducing the energy cost drastically. Interestingly, the loss of these contacts had little or no impact on a common DTN routing protocol. However, neighbor discovery is only the first step towards efficient proximity-based peer-to-peer communication. As future work, we would like to take the next step which is integration of neighbor discovery with data transmission.

## 7. REFERENCES

- [1] ns2 network simulator. <http://www.isi.edu/nsnam/ns/>.
- [2] Opportunistic network environment ONE. <http://www.netlab.tkk.fi/tutkimus/dtn/theone/>.
- [3] G. Ananthanarayanan and I. Stoica. Blue-fi: enhancing wi-fi performance using bluetooth signals. In *Mobisys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 249–262, New York, NY, USA, 2009. ACM.
- [4] P. Bahl, A. Adya, J. Padhye, and A. Walman. Reconsidering wireless systems with multiple radios. *SIGCOMM Comput. Commun. Rev.*, 34:39–46, October 2004.
- [5] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi. ns2-miracle: a modular framework for multi-technology and cross-layer support in network simulator 2. In *Proceedings of ValueTools 2007*, pages 16:1–16:8, ICST, Brussels, Belgium, Belgium, 2007. ICST.
- [6] W. He, Y. Huang, K. Nahrstedt, and B. Wu. Message propagation in ad-hoc-based proximity mobile social networks. In *Proceedings of Pervasive Computing and Communications Workshops 2010*, pages 141–146, 29 2010-april 2 2010.

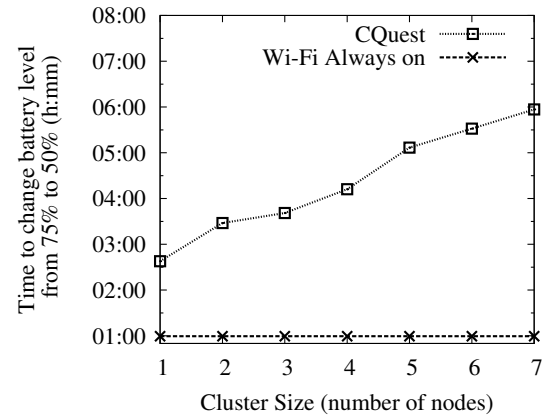


Figure 6: Effect of Cluster Size on Energy

- [7] W. jen Hsu and A. Helmy. On nodal encounter patterns in wireless lan traces. *IEEE Transactions on Mobile Computing*, 9:1563–1577, November 2010.
- [8] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar. U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of IPSN 2010*, pages 350–361, 2010.
- [9] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *Proceedings of MobiCom 2005*, pages 243–257, New York, NY, USA, 2005. ACM.
- [10] M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, REALMAN '06*, pages 31–38, New York, NY, USA, 2006. ACM.
- [11] A. K. Pietiläinen, E. Oliver, J. Lebrun, G. Varghese, and C. Diot. MobiClique: middleware for mobile social networking. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 49–54. ACM, August 2009.
- [12] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31), Jan 2006.
- [13] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of the 8th MobiCom*, pages 160–171, NY, USA, 2002. ACM.
- [14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proc. SIGCOMM Workshop on Delay-tolerant networking*, 2005.
- [15] L. Vu, D. Quang, and K. Nahrstedt. Lessons learned from bluetooth/wifi scanning deployment in university campus. Technical report, Department of Computer Science, University of Illinois, 2010.
- [16] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li. E-smalltalker: A distributed mobile system for social networking in physical proximity. In *Proceedings of ICDCS 2010*, pages 468–477, June 2010.