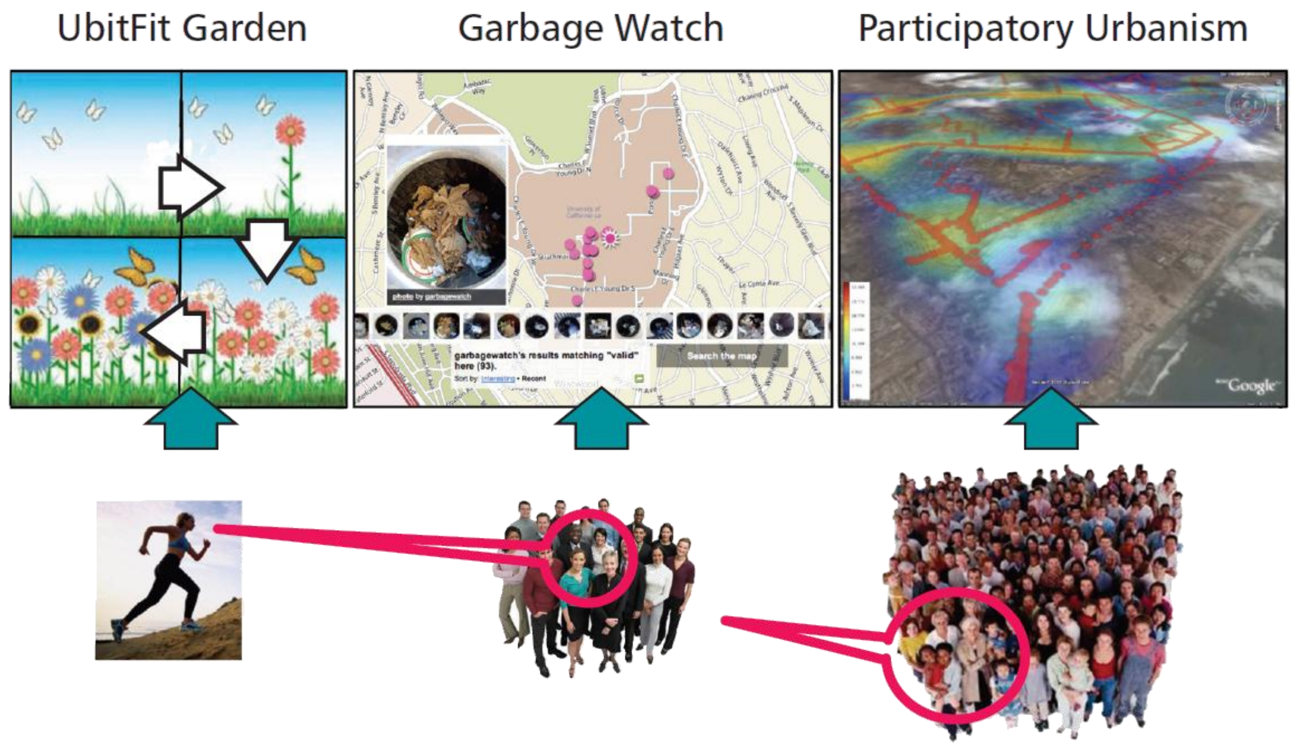
dQuorum: Neighbor Discovery With Distributed Quorum System

Desheng Zhang[1], Tian He[1], Yunhuai Liu[2], Yu Gu[3], Fan Ye[4], Raghu k Ganti[4] and Hui Lei[4]

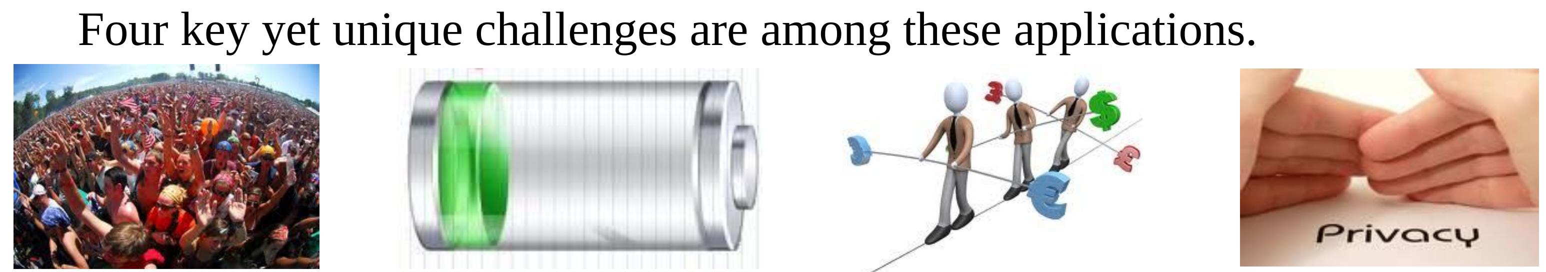
[1]*University of Minnesota, Twin Cities, USA,*
[2]*Shenzhen Institute of Advanced Technology, Shenzhen, China*
[3]*Singapore University of Technology and Design, Singapore*
[4]*IBM T.J. Watson Research Center, Hawthorn, NY, USA*

Introduction

Personal mobile devices, *e.g.*, smartphones, have become extremely popular recently. These devices are equipped with various sensing and wireless modalities, enabling numerous novel applications [1].



Four key yet unique challenges are among these applications.



Dynamic population | Function Availability | Incentive Mechanism | Privacy Preservation

Addressing the dynamic population issue in this poster.

- Among above mobile applications, a fast discovery of highly dynamic neighbor devices is typically required.
- Due to the constant mobility of users, we would typically favor low discovery latency with reasonable accuracy.
- Current Quorum-based neighbor discovery, *e.g.*, GQS [2], is tightly based on the unrealistic assumption in a fully distributed asynchronous network with devices working autonomously, *i.e.*, duty cycles of devices fit some specific prefixed patterns to ensure discovery among any two of devices
- To address the above issue, in this poster, we propose, *dQuorum*, a neighbor discovery protocol with distributed quorum system, accelerating the discovery and reduce the energy consumption with a two-phases discovery process, where the devices leverage the passive assistance from their neighbors to achieve a fast discovery with reasonable accuracy.

Preliminaries

- Quorum system is a superset where any of two subsets of quorums at least have one common element.
- In Quorum-based discovery, time is divided into m×m continuous slots (*quorums*) as a matrix and each device selects one row and one column slots (subset of quorums) to becoming active. Consequently, regardless which row and column a device are selected, it is guaranteed to have at least two common active time slots (common elements) with other devices. A example of GQS [2] is shown in Fig.1
  - ✓ With global parameter m=6, a device *i* selects slot 5, 11, 17, 23, 29, 35 to send its messages to notify its existence, and selects slot 13 to 18 to listen the messages to learn its neighbors' existences.
  - ✓ A device *j* selects slot 3, 9, 15, 21, 27, 33 to send, and selects slot 25 to 30 to listen.
  - ✓ Consequently, device *i* and *j* discover each other at slot 15 and 29, respectively.

			j		*i*	
	1	2	3	4	5	6
	7	8	9	10	11	12
i	13	14	15	16	17	18
	19	20	21	22	23	24
j	25	26	27	28	29	30
	31	32	33	34	35	36

Inactive | *i* *j* Send | *i* *j* Listen | Discovery

Fig. 1 Centralized Quorum System

- The drawbacks of above centralized quorum are as follows.
  - ✓ Global *m*=6: leading to the same duty cycle for all, $(m+m-1)/m^2$.
  - ✓ Continuous listening *m*: causing a significant energy consumption.
  - ✓ Ineffective Discovery: Device *i* and *j* are only aware of existences of each other. The neighbors of they discovered are not shared with each other.

dQuorum Design

- To address above drawbacks, we propose a distributed quorum-based neighbor discovery, *dQuorum*, which works by following virtues.
  - ✓ enabling devices selecting duty cycles based on their own energy budgets,
  - ✓ employing a two-phase scheme to avoid continuous listening,
  - ✓ leveraging outcomes of the neighbor discovery of the detector's neighbors, by letting devices broadcasting neighbor tables in active slots. And, other devices will passively assist *S*, by *distributedly* discovering some of *S*'s neighbors, compared with *S centralizedly* discovering all neighbors in centralized quorum discovery.
- To achieve the similar effect of continuous listening of *m* slots, *S* finds a subset *N*, where the union of active slots of devices in *N* is continuous *m* slots.
  - ✓ *Phase I: Distributed Neighbor Discovery:* Upon the discovery of the continuous *m* slots, the union of neighbor tables of devices in *N* may contain all or a large portion of *S'*s neighbors.
  - ✓ *Phase II*: *Centralized Discovery Collection:* After Phase I, *S* achieves the similar discovery effect by transparently collecting discovery outcomes of all devices in *N* with another maximal continuous *m* slots.

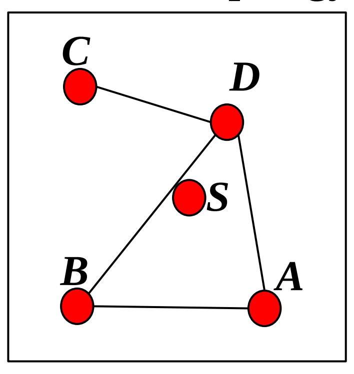
- ✓ Active slots of four discovered neighbors of *S* is shown in Fig.2 decided by their own duty cycles, and the phase length is *m*=5 determined by minimal duty cycle is 1/6.
- ✓ Instead of becoming active for 5 slots from slot 0 to 4, *S* becomes active in slot 3 of Phase I to organically wake up and slot 6 of Phase II to collect discovery outcomes that its neighbor *A*, *B* and *C* obtained in Phase I, achieving similar neighbor discovery effect to 5 active slots of *S*, and yet conserve 60% of energy by reducing active slot from 5 to 2.
- ✓ *S*'s neighbors are unaware of this two-phase transparent operation. The latency is still bounded since *S* will be also becoming active in its original active slots in Phase I and II.

Global Time

Phase I | Phase II

0	1	2	3	4	5	6	7	8	9
			S				*S*		
A		*A*		*A*		*A*		*A*	
B			*B*			*B*			*B*
	C					*C*			
D					*D*				

Network Topology



Neighbor Table of *S*

ID	Neighbors
A	*B, D*
B	*A, D*
C	*D*
D	*A, B, C*

Fig. 2 Distributed Quorum System

- How to select *N* is the essential step to dQuorum. Intuitively, to discover more neighbors and reduce active slots, *S* should select *N* with neighbors who
  - ✓ discover the same or extremely similar neighbor set of S in Phase I;
  - ✓ minimize the discovery outcomes collecting slots of S in Phase II.
  - ✓ enable good tradeoff of our design, *i.e.*, to cover the more slots in Phase I and to avoid cause a prolonged collection process in Phase II.
- To focus energy consumption, we merge all devices becoming active in a slot *i* of Phase II into a neighbor set M_i. *S* should favor *M*s that have active neighbors exhibiting both *temporal diversity* and *spatial similarity* to *S*.
  - ✓ ***Temporal diversity*** is computed as ratio between the number of non-overlapping active slots between *S* and devices in *M* during Phase I, *e.g.*, in Fig.2, temporal diversity between M_5 and S is 1/5.
  - ✓ ***Spatial Similarity*** is computed as ratio between the number of common known neighbors of *S* and devices in *M*, and the total number of known neighbors of *S*, *e.g.*, in Fig.2, spatial similarity between M_5 and S is 4/4.
- Discovery problem can be transformed as: to find a subset *N* consisting of *M*s with the minimal number such that S and *N's* temporal diversity × spatial similarity=1, e.g., M_6 is one of *N*s.
  - ✓ However, this optimized problem can be reduced to Set Cover problem, which is NP-complete and requires exponential solutions[2].
  - ✓ In this paper, we employ a greedy scheme to add the *M* with the largest temporal diversity × spatial similarity into *N* one by one, until S and *N's* temporal diversity × spatial similarity=1.

Implementation & Conclusion

- To evaluate the performance of dQuorum in a real world setting, we implemented dQuorum on the TinyOS/Mote platform. During the testbed experiments, we deploy 10 TelosB sensor devices and utilize a mobile toy car attached with another TelosB as the discovering device. The testbed setup is shown in Fig.3.

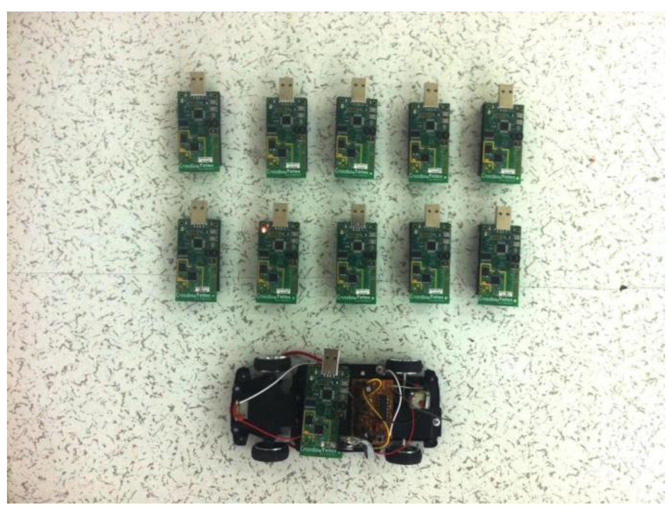


Fig. 3 Testbed Setup

- Fig.4 plots the testbed experiment results on the CDF of discovery latency with *m*=25, where our design dQuorum exhibits significantly reduction in discovery latency, compared to centralized quorum-based neighbor discovery protocol, *e.g.*, for discovering 80% of neighbor devices, dQuorum and centralized quorum-based neighbor discovery protocol spends 12.7s and 31.6s, respectively, achieving a improvement of 59.8%.

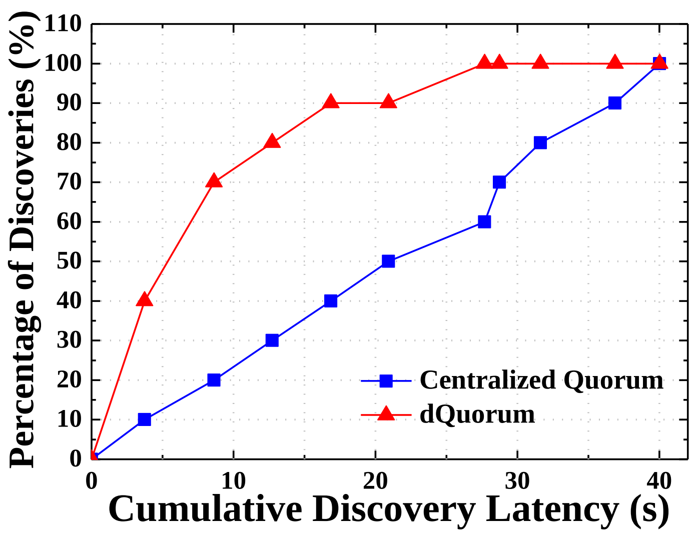


Fig. 4 Discovery Latency CDF

- Fig5. shows the impact of three different duty cycles on the average discovery latency of one device, where dQuorum always outperforms centralized quorum-based neighbor discovery protocol by 63% at most. As the duty cycle increases, the performance gap also increases from 33% to 63%.

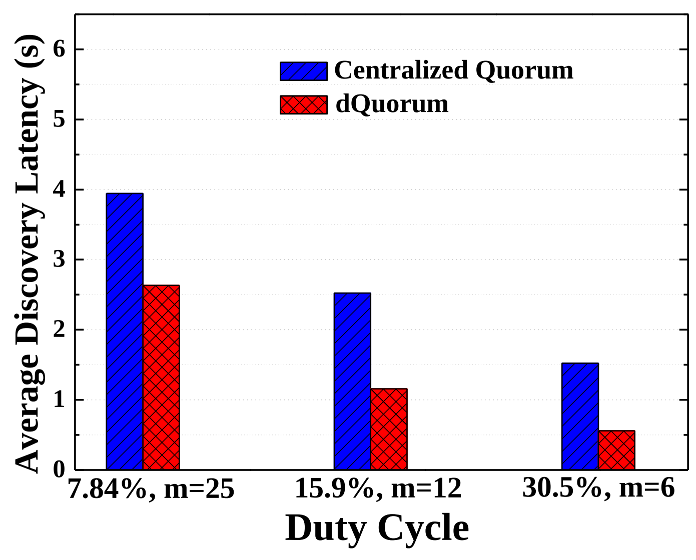


Fig. 5 Average Discovery Latency

- In this poster, we propose a neighbor discovery protocol, utilizing a two-phase neighbor discovery with temporal diversity and spacial similarity among devices in mind. The testbed experimental results indicate dQuorum with two-phase neighbor discovery is more effective than centralized quorum-based schemes to accelerate discovery.

References

1. Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. 2010. A survey of mobile phone sensing. Comm. Mag. 48, 9 (September 2010), 140-150.
2. J. Burke et al. Participatory sensing. Workshop on World-Sensor-Web, co-located with ACM SenSys, 2006.
3. S. Lai, B. Ravindran, and H. Cho. Heterogenous quorum-based wake-up scheduling in wireless sensor networks. Computers ,IEEE Transactions, 2010.