Blockchain-based IoT Secured Energy-Efficient Collaborative Neighbor Discovery Protocol

Dan Ye

Abstract- This paper proposes an Energy-efficient Secured Distributed Collaborative Neighbor Discovery protocol for IoT mobile sensing applications to enhance on IoT capabilities and efficiencies. Maximum throughput access control model with ultra-low-power constraints is proposed for blockchain-based IoT framework. Simulation result demonstrates the proposed protocol can achieve better discovery performance with minimum discovery latency and maximum duty cycle. Analysis of the comparison results present best choice of primes and duty cycle in the designed low-complexity algorithm.

Index Terms-LTE IoT, neighbor discovery, blockchain, security.

1. Introduction

NailO transmits the sensor data to a mobile phone or smart device, fingernails as an input surface, enabling wearers to customize the device to fit the wearer's personal style. NailO allows wearers to perform different functions on a phone or PC with different gestures.

In mobile networking application, wearable [1], [2] devices carry various types of sensors and interact with neighbor devices to exchange sensing data.

Neighbor discovery—bootstrapping primitive that discovers all the neighbors of a mobile device, supporting topology and clustering, medium access control, routing and other basic networking functionality. An efficient neighbor discovery protocol should enable a node to discover its neighbors with constrained delay for other proximity-based applications interacting with nearby peers in real time.

Devising effective neighbor discovery protocols for distributed IoT sensing applications require a strigent ultra-low-power constraints [3]. Limiting neighbor discovery delay is one of design challenges. Minimizing neighbor discovery delay and maximizing energy conservation is the main design objectives in neighbor discovery protocols.

Dr.Dan Ye is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 10617 Taiwan.

Security mechanism along with localization procedure without any additional overhead is also one of important design requirements. Novel secure routing protocol in IoT system is proposed to implement message confidentiality and integrity for robustness, reliability and trustworthiness.

This paper proposes an energy-efficient secured distributed collaborative neighbor discovery protocol in asynchronous manner, that allows multiple nodes to operate at low duty cycles and discover and communicate with one another during opportunistic encounters[4]. The proposed low-power, asynchronous neighbor discovery protocol in this paper is co-deployed on two systems to discover each other. The key challenge is to achieve optimization balance between low-power operation, maximize lifetime and active vigilance to detect the emergence of new links.

The promising feature of proposed neighbor discovery protocol should support 1 million nodes per km IoT devices density. It should last for more than 10 years of battery life with capacity 10 Tbps per km limiting 1 ms latency.

The remainder of this paper is structured as follows: Section 2 proposes energy-efficient secured distributed collaborative neighbor discovery protocol in combination with graph theory. Section 3 presents the details on security mechanism. Section 4 proposes a maximum throughput access control model under ultra-low-power constraints for blockchain-based IoT framework. Section 5 presents the main theoretical result. Section 6 evaluates energy-efficient secured distributed collaborative neighbor discovery protocol numerically via simulations, then discuss the experimental implementation and evaluation. A summary concludes the paper in Section 7.

2. Energy-efficient secured distributed collaborative neighbor discovery protocol

Efficient path prediction algorithm can assist design fast, secure, reliable, flexible predictable neighbor discovery protocol [5]. This protocol [6] ensures that there are no overlapping radio on time.

This section proposes low power maximum throughput energy-constrained hybid synchronous and asynchronous rendezvous automatic neighbor discovery protocol. Cooperative co-located discovery under low duty cycles is to maximize lifetime.

New protocol designs that global counter is divisible by selected primes. Low complexity, adaptation distributed neighbor discovery protocol. Every IoT mobile sensing node chooses discovery latency or a desired duty cycle firstly. Global coordination of duty cycles minimizes discovery latency. The global counter in the novel protocol can automatically selects primes that match the desired duty cycle or discovery latency at each multiple of the chosen primes. Nodes can be assigned to different clusters such that inter-cluster discovery times is much faster than without classification, nodes can adjust duty cycles to minimize a certain discovery latency. New protocol proposes clustering pattern, that a group of nodes move together as a unit. A node periodically beacons its neighbor presence, and flexiblely chooses best node to be a neighbor instead of first node. A node only beacons and listens neighbors after adjustable period. Nodes maintain time synchronization to discover one or more neighbor by sending long preambles. Varying energy availability, idle listening dominates the system power budget, balances power supply and reduces the listen duty cycle.

Asymmetric duty cycles is used for low-power listen and sense that must be adjusted to the available energy, and adaptive listen period.

2.1 Protocol model formulation-simplified pattern

Two nodes m and n, pick two numbers, A_m and A_n, are relatively prime and $1/A_m$ and $1/A_n$ are equal to m and n's desired duty cycles. Time is divided into consecutive periods. Nodes m and n starts counting periods at times a_m and a_n, with counters C_m and C_n, with m and n counts synchronized to reference period. If $C_m|A_m = 1$ (C_m is divisible by A_m), m turns on its radio for one period and beacons. When $C_n|A_n = 1$, both m and n turn on their radios during the same period, they can exchange beacons and discover each other.

Assume one overlapping period every $A=A_mA_n$ periods, letting K represent the reference period number,

$C_m = K - a_m$

$C_n = K - a_n$

The main objective is to search K such that $C_m|A_m$ and $C_n|A_n$.

$K \equiv a_m \pmod{A_m}$

$K \equiv a_n \pmod{A_n}$.

If k_0 is one solution, $K=k_0+zA$ for some integer z.

Where $k_0=a_mb_mA_m+a_nb_nA_n$,

$b_mA_n \equiv 1 \pmod{A_m}$

$b_nA_m \equiv 1 \pmod{A_n}$.

If node m select $A_m=3$, so m's duty cycle is DC=33%, start counting at reference period K=1, so that $a_m = 1$, with counter values $C_m=0$. Let node n select $A_n=5$, so n's duty cycle is DC=20%, start counting at reference period K=2, so $a_n=2$, with counter values $C_n=3$. Both m and n have overlapping on slots, they can communicate.When K=7 and K=22, both m and n are turned on and can discover each other. Discovery procedure is demonstrated in Figure 1.

$K \equiv 1 \pmod 3$

$K \equiv 2 \pmod 5$

When K=7,

(7-1)|3

(7-2)|5

An analytic solution requires finding b_m and b_n

$5b_m \equiv 1 \pmod 3$

$3b_n \equiv 1 \pmod 5$

Values of $b_m=2$ and $b_n=2$, one solution k_0 is

$k_0=a_mb_mA_n+a_nb_nA_m$

$k_0=1*2*5+2*2*3$

$k_0=22$

All solutions are unique (mod 15),

$k_0=22 \pmod{15}=7$

So $K=7+15z$, for all $z \in \mathbb{Z}^+$.

2.2 Cluster pattern

In simplified discovery pattern, nodes m and n are discovered twice each other. Synchronized their counting with the reference phase, this section proposes new cluster discovery pattern with $A_n=2$, DC=50%. So nodes m and n are discovered at times K=4, K=10, K=16 and K=22. Considering minimizing energy overhead, maximum throughput with ultra-low-power constraints, K=4+6z for all $z \in \mathbb{Z}^+$. New discovery protocol in cluster pattern is presented in Figure 2.

2.3 Choosing best duty cycle

Choosing best appropriate moduli, so best satisfy their individual duty cycle requirements, when $A_m=A_n$, nodes m

K	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C_m	-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
C_n	-	-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Fig 1. Discovery timeline. Two nodes, m and n, start their counters, C_m and C_n, at times K=1 and K=2, with periods A_m=3 and A_n=5, and duty cycles of 33% and 20%. The colored cells indicate times when the nodes m and n turn on their radio. Both nodes are discovered each other at times K=7 and K=22. This simple pattern repeats when K=7+15z, for all $z \in \mathbb{Z}^+$.

K	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C_m	-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
C_n	-	-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Fig 2. New Discovery timeline. Two nodes, m and n, start their counters, C_m and C_n, at times K=1 and K=2, with periods A_m=3 and A_n=2, and duty cycles of 33% and 50%. The colored cells indicate times when the nodes m and n turn on their radio. Both nodes are discovered each other at times K=4, K=10, K=16 and K=22. This cluster pattern repeats when K=4+6z, for all $z \in \mathbb{Z}^+$.

and n may never discover each if they wake up with the same period but different phase.

Define node m selects the best duty cycle with two primes D_{m1} and D_{m2} ($D_{m1} \neq D_{m2}$), thus the sum of their reciprocals equals to the desired duty cycle

$$DC_{best} \approx \frac{1}{D_{m1}} + \frac{1}{D_{m2}}.$$

Duty cycles are independently selected at different nodes, for every pair of nodes m and n. Let intra-node pairs are coprime

$\gcd(D_{m1}, D_{m2})=\gcd(30,77)=1$

$\gcd(D_{n1}, D_{n2})=\gcd(35,66)=1$

However, inter-node pairs are not coprime, discovery may not success. Node m starts counting at times K=30z and K=77z, for all $z \in \mathbb{Z}^+$. Node n starts counting at times K=35z+1 and K=66z+1, for all $z \in \mathbb{Z}^+$.

$\gcd(D_{m1}, D_{n1})=\gcd(30,35)=5$

$\gcd(D_{m1}, D_{n2})=\gcd(30,66)=6$

$\gcd(D_{m2}, D_{n1})=\gcd(77,35)=7$

$\gcd(D_{m2}, D_{n2})=\gcd(77,66)=11$

2.4 Choosing best primes

The choice of primes can determine discovery latency. A good choice can result in low discovery latency. Randomize the choice of prime pairs to reduce the chance that two nodes picked the same pair if they both select the same duty cycle.

If nodes can be assigned to different clusters, member of a cluster requires the good choice of prime pairs. For each duty cycle, every node generates an ordered list of prime pairs that can satisfy the duty cycle.

A node chooses at random one of the prime pairs assigned to its cluster. The new cluster discovery protocol ensures that nodes in different clusters are assigned distinct pairs, so that discovery latency can be greatly improved. Good choice of inter-cluster pairs is important for cluster label assignment. Maximize the likelihood that overlapping slots result in discovery, a beacon is transmitted at the beginning and end of a slot.

Blindly transmitting when the channel is busy. Effective cutting off the timing of slot, could alleviate long delay.

2.5 Discovery latency

Duty cycle or beacon rate is computed to satisfy discovery latency. The objective is to compute minimum duty cycle and discovery latency. The main task is to convert maximum discovery latency T_{max} to duty cycle. Discovery latency is limited by assigning different nodes to different clusters. Two nodes operating with primes D_{m1} and D_{m2} will discover each other at most $D_{m1}D_{n2}$ counter periods, where each counter period is of length T_{slot}.

$$D_{m1}D_{n2}T_{slot} \leq T_{max} \quad (1)$$

$$D_{m1}=D_{n2}=D \leq \sqrt{\frac{T_{max}}{T_{slot}}} \quad (2)$$

Minimum duty cycle satisfies

$$DC_{min} \geq \frac{1}{D}+\frac{1}{D}=\frac{2}{D}. \quad (3)$$

Minimum beacon rate is obtained by

$$R_{beacon} \geq \frac{2}{DT_{slot}}=\frac{2}{\sqrt{T_{max}T_{slot}}}Hz \quad (4)$$

Although the beacon rate increases with smaller T_{slot} values, effective duty cycle decreases

$$DC \geq \frac{2}{D}=2\sqrt{\frac{T_{slot}}{T_{max}}} \quad (5)$$

2.6 Duty Cycle Granularity

D_{m1}	1	2	2	2	2	2	2	3
D_{m2}	0	3	5	7	9	11	13	5
DC(%)	100	83.3	70	64.3	61.1	59	57.6	53.3

Fig 3. Best duty cycles (DC)

3. Security mechanism

Neighborhood discovery protocol identifies actual neighbors by position-based routing distributed protocol that selects appropriate neighbor to forward data via multipoint relay routing scheme.

Partial ND protocol fails to discover and verify all neighbors[7]. This problem can be solved by autonomous device coordination via Peer to Peer (P2P) messaging and distributed file sharing.

In the P2P messaging approach, encrypted messaging and transport maintaining low latency with guaranteed delivery store and forwarding of messages with other connected devices. Such messaging capabilities can be achieved using structured P2P networks where overlay topology and protocol that any node can efficiently search the network for another peer. Distributed hash table (DHT) can be used to enable peers to search for other peers on the network using a hash table with (key, value) pairs stored in the DHT. Each end point would generate its own unique public-key based address (a hashname) to send and receive encrypted packets with other end points and any participating node can efficiently retrieve the value associated with a given key.

Distributed file sharing enables decentralized software updates, device based analytics reporting secure file and data sharing, large orders of magnitude. Such transfers can be achieved by means of distributed P2P networks using DHT. Integrating the messaging layer with the blockchain layer, not only for interoperability, but also to capture critical trusted communcations for future verification. In peer to peer file sharing protocol for decentralized peer to peer IoT solution, multiple clients adopt different NAT systems to maintain effective communication.

4. Maximum throughput access control protocol with ultra-low-power constraints IoT blockchain scheme

This section proposes a new framework for maximum throughput access control with ultra-low-power constraints in IoT based on the blockchain technology. [3] provides efficient methods to obtain maximum groupput and anyput among a set of energy constrainted nodes with heterogeneous power budgets.

4.1 Register a new resource

A private key A.sk is a number, used to create signatures for both transaction authentication and integrity that are required to prove ownership of resources and control access with addresses extracted from its corresponding public keys in Maximum Throughput Access transaction, whereas the public key is used to identify a particular recipient.

A parent key can derive a sequence of children keys, each of which can derive a sequence of grandchildren keys. Resource's addresses are extracted from a child public key corresponding to its RO. RO is able to identify resources by extracting address from the generated keys. RO has ability to derive public child keys from public parent keys, without private keys. RO manages maximum throughput access control policies with ultra-low-power constraints for all the registered resources by deriving the corresponding private keys to sign transactions.

Those resources can use public key derivation function to create a new address for every transaction. Those interacting entitles could obtain a public key to get an address for receiving access request without private keys.

R/S RO can control the access to all the resources associated to generated addresses by one private key which is seed key.

RO knows the address of the requester

1. RO defines for Resource Address rs, Requester Address rq an access control policy $POLICY_{rs,rq}$.

2. Bitcoin transforms access control policy to a scripting language and generates a maximum throughput access Transaction with ultra-low-power constraints, that will be signed with RO private key then propagated to the network

$$\text{POLICY}_{rs,rq} \rightarrow \pi_x \quad (6)$$

Transactions

$$Tx=(m,sig_{rs}(m)) \quad (7)$$

Where m=(IDx, Vin[input1(tokenbase,rs)],Vout[output1(rq, π_x , $TKN_{rq,rs}$)])

Vin[]: Each input in the vector inputs. Vout[]: Each output in the output vector consists of two parts, TKN (access token and requester address) and locking script.

IDx: the index of current transaction identifier Tx where x=H(Tx).

TKN: encrypted access token is the value of the transaction, with public key extracted from rq address, recorded in the blockchain.

rs: the address of requested resource.

rq: the address of requester who is the receiver of the current transaction Tx.

π_x : locking script, access control policy in scripting language.

3. Each node verifies the transaction within the transaction validation process.

4. If the transaction is valid output: $TKN_{rq,rs}$ is recorded in the blockchain.

5. The requester scans TKN database, by scanning the blockchain and collecting all TKN associated to the client address. If TKN database contains a TKN associated to resource, generates a maximum throughput under ultra-low-power access transaction, then sends a request to owner containing the Address.

$$\text{ScanTKN(rq)} \rightarrow TKN_{rq,rs} \quad (8)$$

$$\text{decrypt}(TKN_{rq,rs}) \quad (9)$$

$$\text{GetLockingscript(TKN)} \rightarrow \pi'_x \quad (10)$$

Where π'_x is the locking script in the corresponding maximum throughput under low power access transaction.

6. The requester fulfills maximum throughput under low power access control condition in π'_x and generates an unlocking script

$$\text{MeetAccessControlPolicy}(\pi'_x) \rightarrow \psi \quad (11)$$

7. Maximum throughput under low power access transaction Tx=(IDx,Vin[input1(ref,rs,ψ)],Vout[output1(rq,$TKN_{rq,rs}$)])

ref: point on the previous outp where a $TKN_{B.pk,rs}$

ψ : unlocking script that has meet to get $TKN_{B.pk,rs}$

c.pk: address of one of resources

π_x : a new locking script access control policy

$TKN_{c.pk,rs}$: encrypted access token.

This transaction to blockchain enables the delivery of encrypted access token $TKN_{rq,rs}$ to the requester.

8. Bitcoin broadcasts the transaction to the network

9. Network verifies and validates the transaction within the validation protocol if it was valid it will be included in the blockchain, and a notification is sent to the sender.

10. Once the transaction appears in the blockchain meaning that the network witnesses that the client has met the access condition (unlocking script) then TKN could be delivered to client.

11. The requester sends a remote access using a token distributed by local blockchain.

12. Blockchain and bitcoin network

All types of bitcoin network nodes are running on the local machine. Bitcoinj is chosen as implementation of bitcoin protocol. Send/receive transactions without a local copy of bitcoin core.

13. Access control policy is based on maximum throughput and ultra-low-power. RO keeps control over token for each use.

14. To transfer the token encrypted in the transaction, bitcoin provides ability to store data in the blockchain. Access token is encapsulated inside the output script.

15. Both IoT and non-IoT resources are identified with bitcoin addresses.

16. RO implemented on maximum throughput under ultra-low-power access control policy.

17. Token was encapsulated in the transactions.

Bitcoin sends maximum throughput under ultra-low-power access transaction.

18. Blockchain and bitcoin network validates the transactions.

19. The requester scans with access tokens delivered.

20. The requester fulfills access control condition to obtain the TKN through maximum throughput under ultra-low-power access transaction.

21. The requester obtains the TKN.

Fig.4. Maximum throughput access control protocol with ultra-low-power constraints IoT blockchain scheme

5. Numerical results

This section evaluates the throughput and latency performance of energy-efficient secured distributed collaborative neighbor discovery protocol operating in groupput and anyput modes.

According to 3GPP specification about 5G/NR frame structure [9], channel bandwidth up to 100 MHz per NR carrier, maximum channel bandwidth per NR carrier is 400 MHz. Maximum number of subcarrier is 3300 or 6600 in Rel 15. Subframe Duration is fixed to be 1 ms. Frame length is fixed to be 10 ms. Subcarrier spacing is from 15KHz to 480 KHz. Number of subcarrier per PRB is 12. TDD and FDD wideband and narrow band, sub 6GHz and mmWave. 1 ms end-to-end radio latency.

Homogeneous networks consist of nodes with the same power budget and consumption levels, i.e., P_m=P, L_m=L, X_m=X, $\forall m \in M$. A large value of M results in lower latency [10], since every node is more likely to receive when more nodes exist in the network.

Energy harvesting tags [11] need to reply on the power that can be harvested from sources such as indoor-light or kinetic energy, which provide 0.01-0.1 mW[12],[13],[14]. Power budget usually at the order of 1-10 mW. In Panda [15], each node does not need to know the number of nodes in the network, M, and the power budgets [16] and consumption levels of other nodes. Most of neighbor discovery protocol [17] do not consider different listen and transmit power consumption levels of the nodes, or do not account for different power budgets.

Fig.5 indicates slot length varies with increasing discovery latency under different duty cycles. As the duty cycle rises, slot length augments significantly. Slot length of simple pattern is shorter than cluster pattern. Duty cycle of cluster pattern is larger than simple pattern.

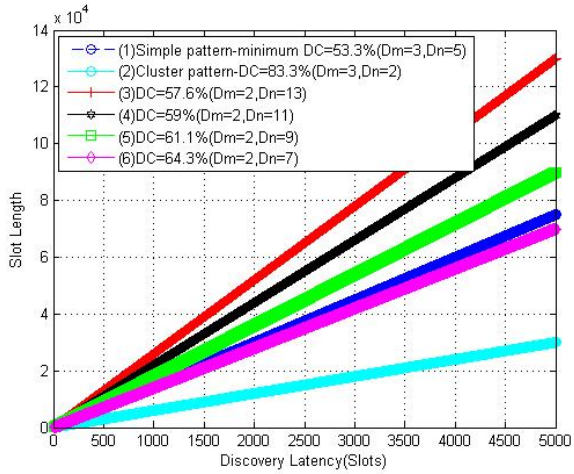

Fig.5. Slot length comparisons under different discovery latency with distinctive duty cycles

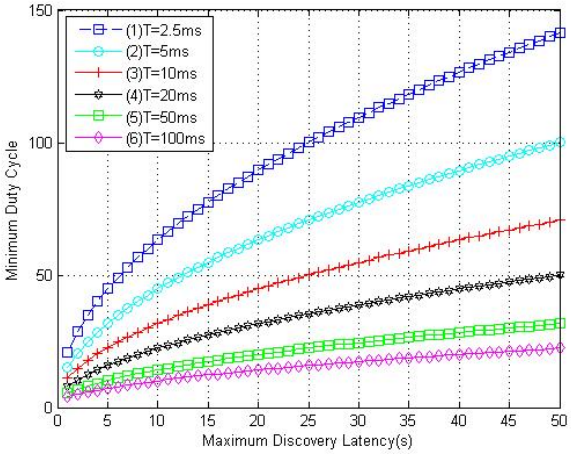

Fig.6. Minimum duty cycle vs. maximum discovery latency with different time lengths under cluster pattern K=4+6z.

Fig.6 demonstrates minimum duty cycle variations with soaring maximum discovery latency. The shorter time length, the larger minimum duty cycle, the longer maximum latency.

6. Experimental evaluation

In this section, first evaluate energy measurements [18] performed on the IoT nodes, the method by which nodes can estimate the number of neighbor discovery nodes. Then, we evaluate the method by which nodes can estimate the density of neighbor nodes[19]. Finally, we experientally evaluate the performance of latency[20].

Fig.7 plots the relationship between discovery rate and time slot. Average discovery rate scales linearly with slot length. Under 10 ms slot value, 200 discoveries are observed within average discovery period of 24 seconds.

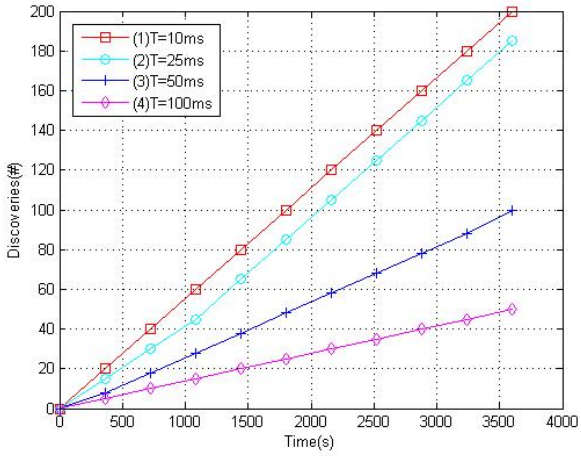

Fig.7. Discovery timeline for different values using best primes and symmetric pairs.

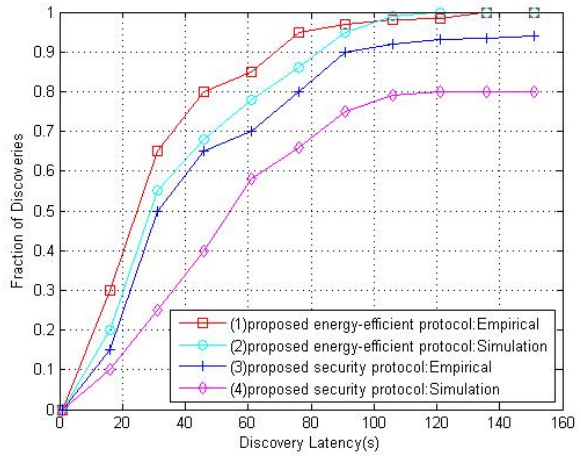

Fig.8. The nodes are operating with a 10 ms slot length and a 2% duty cycle using best primes and symmetric pairs.

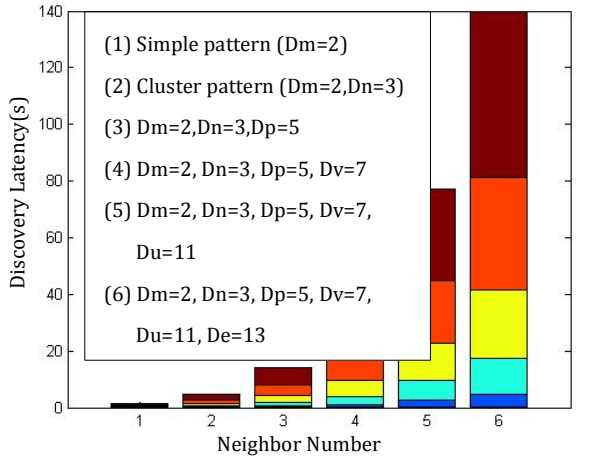

Fig.9. The discovery latency of six neighbors when a node joins a cluster with seven nodes. The nodes are operating with different slot length marked with distinct colors. Black: T_{slot}=1ms; Blue: T_{slot}=10 ms; Light blue: T_{slot}=20 ms; Yellow: T_{slot}=30 ms; Red: T_{slot}=40 ms; Dark Red: T_{slot}=50ms.

For 25 ms or longer slot value, maximum 105 discoveries are observed within average discovery period of 60 seconds.

Fig.8 illustrates simulated and empirical results analyzed in cluster pattern. The unique neighbor discoveries of one particular listening node were recorded. The discovery latency of each neighbor was recorded. The empirical discovery latency is lower than the simulated discovery latency.

Fig.9 elaborates discovery latency of six neighbors when a node enters a cluster with seven nodes. The more neighbor number of nodes, the longer discovery latency. According to different colors, the longer slot length lasts, the longer discovery latency elapses.

7. Conclusion

This paper develops distributed neighbor discovery protocol that control the nodes' discovery latency. The proposed cluster pattern neighbor discovery protocol achieves discovery latency shorter than other discovery protocols and its security level is higher than other discovery protocols. Furthermore, energy-efficient collaborative machanism can improve discovery performance. The key application of this discovery protocol is integrated into sensing IoT mobile terminals.

Reference

[1]. Hsin-Liu Kao, Deborah Ajilo, Oksana Anilionyte, Artem Dementyev, Inrak Choi, Sean Follmer, Chris Schmandt, "Exploring Interactions and Perceptions of Kinetic Wearables", DIS '17, June 10-14, 2017, Edinburgh, United Kingdom.

[2]. Artem Dementyev, Hsin-Liu Kao, Inrak Choi, Deborah Ajilo, Maggie Xu, Joseph A. Paradiso, Chris Schmandt, Sean Follmer, "Rovables: Miniature On-Body Robots as Mobile Wearables", UIST'16, October 16–October 19, 2016, Tokyo, Japan.

[3]. Tingjun Chen, Javad Ghaderi, Dan Rubenstein, and Gil Zussman, "Maximizing broadcast throughput under ultra-low-power constraints," 26 Apr, 2017.

[4]. Prabal Dutta, David Culler, "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications" , SenSys'08, November 5–7, 2008, Raleigh, North Carolina, USA.

[5]. S. Vasudevan, M. Adler, D. Goeckel, and D. Towsley, "Efficient algorithms for neighbor discovery in wireless

networks," IEEE/ACM Trans. Netw., vol. 21, no. 1, pp. 69–83, 2013.

[6]. W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," IEEE Commun. Surveys Tuts., vol. 16, no. 3, pp. 1448–1459, 2014.

[7]. Panos Papadimitratos, Marcin Poturalski, Patrick Schaller, Pascal Lafourcade, David Basin, Srdjan Capkun, Jean-Pierre Hubaux, "Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking", IEEE Communications Magazine, 2008.

[8]. Aafaf Ouaddah, Anas Abou Elkalam and Abdellah Ait Ouahman, "FairAccess: a new Blockchain-based access control framework for the Internet of Things", Security and communication networks, 2016; 9:5943-5964, wiley, DOI:10.1002/sec.1748.

[9]http://sharetechnote.com/html/5G/5G_FrameStructure_Candidate.html

[10]. A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in Proc. ACM/IEEE IPSN'10, 2010.

[11]. T. Chen, G. Chen, S. Jain, R. Margolies, G. Grebla, D. Rubenstein, and G. Zussman, "Demo abstract: Power-aware neighbor discovery for energy harvesting things," in Proc. ACM SenSys'16, 2016.

[12]. M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," IEEE Trans. Mobile Computing, vol. 12, no. 9, pp. 1853–1865, Sept. 2013.

[13]. M. Gorlatova, J. Sarik, G. Grebla, M. Cong, I. Kymissis, and G. Zussman, "Movers and shakers: Kinetic energy harvesting for the internet of things," IEEE J. Sel. Areas Commun., vol. 33, no. 8, pp. 1624–1639, 2015.

[14]. S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," IEEE J. Sel. Areas Commun., vol. 33, no. 3, pp. 360– 381, Mar. 2015.

[15]. R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, "Panda: Neighbor discovery on a power harvesting budget," IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 3606–3619, Dec. 2016.

[16]. W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE/ACM Trans. Netw., vol. 12, no. 3, pp. 493–506, 2004.

[17]. P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in Proc. ACM SenSys'08, 2008.

[18]. S. Chen, P. Sinha, N. B. Shroff, and C. Joo, "A simple asymptotically optimal joint energy allocation and routing scheme in rechargeable sensor networks," IEEE/ACM Trans. Netw., vol. 22, no. 4, pp. 1325– 1336, 2014.

[19]. W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in Proc. IEEE INFOCOM'14, 2014.

[20]. L. Chen, R. Fan, K. Bian, M. Gerla, T. Wang, and X. Li, "On heterogeneous neighbor discovery in wireless sensor networks," arXiv preprint arXiv:1411.5415, 2014.