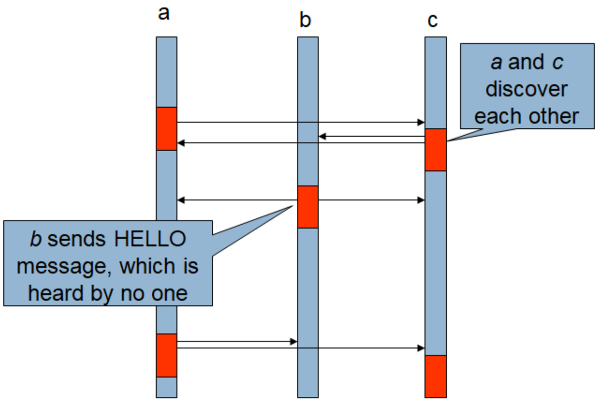

Figure 4. Process of neighbour discovery

As shown in Figure 4, node a can receive HELLO from the node c. That way, they can discover each other. However, the nodes a and c are not aware of the presence of b and vice versa. This indicates the need for neighbour discovery.

2.1 Initial neighbor discovery

A sensor node performs initial discovery when it has no information about its neighbors. Before doing this, the sensor node also cannot communicate with the base station. Therefore it is functioning, and usefulness is limited. Therefore, there is a need for initial neighbor discovery and establish connectivity to the base station. Since it is necessary, the energy usage for discovery is justified as it is one only once. However, when the node is operational, it needs to perform continuous discovery from time to time, as long as the node lives in the network. Therefore optimization of continuous neighbor discovery is crucial for improving the lifetime of the network.

2.2 Continuous Neighbor Discovery

Before performing continuous neighbor discovery, the node is well aware of its immediate neighbors. Therefore this operation is made together with already known neighbors in order to reduce energy consumption. On the contrary, each node has to execute initial neighbor discovery separately. The purpose of continuous neighbor discovery is to detect all neighbor nodes and also find the shortest path data transfer. As the sensor nodes are randomly deployed in some geographical area, initial neighbor discovery is made once when a sensor is deployed. However, continuous neighbor discovery is needed due to the disruption of wireless connectivity and the loss of local synchronization due to clock drifts.

3. Neighbor Discovery Schemes

There are many neighbor discovery approaches found in the literature. This section provides a review of them. The approaches discussed in this section include U-connect [9], Disco [8], SearchLight [42], Hedis [29], Todis [29] and Prime Block Diagram (PBD) [33].

3.1 U-Connect

Kandhalu et al. [9] proposed U-connect, which is a low-latency asynchronous neighbor discovery protocol with energy efficiency. U-connect is designed, and its latency is characterized, and its power consumption is analysed. Then it is evaluated with the power-latency metric. U-connect is believed to be a unified protocol that can address neighbor discovery in two settings. They are called symmetric and asymmetric problems. In the process, two nodes can choose m different prime numbers for discovering neighbors. When nodes use the same pair of prime numbers, a worst-case latency performance is analysed.

In order to have better performance, U-connect characterizes network and neighbor discovery schedules. The latency discovery is associated with the following.

U- ψ u (m, t) = 1, if [t] p = 0 or 0 ≤ [t] p 2 < 0, otherwise p+1 2 (13)

Here the prime number is denoted as p. The case is considered where p>2 for simplicity. The U-connect protocol is designed in such a way that it works well with worst-case latency, which is high. The simulation study revealed that U-connect provide a guarantee for a common active slot. Other advantages of the U-connect include improved latency and energy efficiency.

3.2 Disco Approach

This approach is explored by Dutta and Culler [8]. In WSNs, the low-power systems that are awake at different times need to discover neighbors. In such cases, the nodes need to use their radios at low duty cycles. This is the requirement in order to maximize the lifetime of the WSN. It also needs to be vigilant about the emergence of new links and the disappearance of old links. The two activities are not odds, as vigilance and low-power operations are contradicting. In such networks, Disco is the solution provided to have asynchronous neighbor discovery and solve the problem of rendezvous scheduling. The underlying method in Disco chooses two prime numbers in such a way that their reciprocal's sum is equal to the duty cycle of an application in question. Each node maintains a local counter, and it is incremented when the counter is divisible by one of the prime numbers selected. Then the node turns on the radio for the period of one counter. Disco needs an application to select the desired duty cycle and identify a node class. The Disco selects prime number automatically based on the duty cycle matching, and then radio is turned on at every multiple of the selected prime.

When a node is a wake-up, it can listen or beacon or perform both. Disco performs well in terms of rendezvous frequency, discovery latency, and flexibility for applications. Nodes can achieve discovery latency desired by adjusting duty cycles. The flexibility serves interaction patterns, duty cycles, and different needs of the applications. Talking, docking, and flocking are the three common patterns exhibited by Disco. Discovery or rediscovery of neighbors helps two different nodes with different duty cycles in the presence of a lack of current synchronization to discover each other. In such cases, rendezvous of Disco allows the delivery of messages to previously discovered nodes in controllable and predictable latencies.

3.3 Hedis and Todis

These are the two neighbor discovery protocols proposed by Chen et al. [29]. Hedis stands for heterogeneous discovery as a quorum based protocol while Todis stands for Triple-Odd based discovery as a co-primality based protocol. These two protocols guarantee the process of asynchronous neighbor discovery. They operate in heterogeneous environments with different duty cycles used by each node. The granularity of duty cycles is optimized in order to have better performance. Hedis match actual duty cycles as it is an optimal quorum based approach. According to the design of the Hedis schedule, node a with given duty cycle, the schedule is considered as $sa=\{sta\}0\le t<n(n-1)$, which has n(n-1) time slots. Hedis and Todis are evaluated with different numbers of consecutive odd integers for building a wake-up schedule. There is a requirement of co-prime pair property, which allows a node to choose three consecutive odd integers. From the empirical study, it is understood that both Hedis and Todis can optimize duty cycles in terms of granularity with two approaches for neighbor discovery named quorum based and co-optimality based. Both the protocols can perform well with neighbor discovery latency.

3.4 SearchLight

It is a matrix-based neighbor discovery protocol. It is simple to be built. However, it compromises latency and energy efficiency. It uses a matrix to have neighbor discovery schedules. SearchLight [16] is an asynchronous neighbor discovery protocol. It is built based on three ideas that are basis. It improves periodic awake slots and for probing. It facilitates awake slots to cover a large time window. It can use probabilistic techniques.

3.5 Prime Block Design Based Neighbor Discovery

Lee et al. [33] proposed a neighbor discovery protocol based on the concept of prime block design (PBD). It provides a near-optimal solution for asynchronous wake-up cycles in WSN. It is an extension to its predecessor known as Balanced Incomplete Block Design (BIBD). The PBD works well with both symmetric and asymmetric duty cycles. It adds less number of duty cycles in excess to that of BIBD for performance improvement. Its advantage is that it is more efficient than other protocols. However, it has a particular limitation. It is the lack of availability of BIBD blocks specific duty cycles.

3.6 Other Neighbor Discovery Approaches

Code based approach [1], survey of neighbor discovery protocols [2], [11], [12], [14], [20], [30], [32], block design based protocols [3], energy efficient neighbor discovery [4], asynchronous neighbor discovery in duty cycled networks [5] and neighbor discovery in cognitive radio networks [6] are found in the literature. Other approaches found include a generic flexible protocol [10], smart phone neighbor discovery [13], secure neighbor discovery [15], adaptive neighbor discovery [16], quorum based approach [17], fast neighbor discovery [18], neighbor discovery and link quality estimation [19], low power neighbor discovery [21], broadcast foreigner discovery [22], x-raying neighbor discovery [23], neighbor discovery in 3D scenarios [24], neighbor discovery for security [25], multi-packet reception based neighbor discovery [26], multi-channel neighbor discovery [27] and heterogeneous neighbor discovery [28]. Neighbor discovery for opportunistic networking [31], group-based neighbor discovery [34] and continuous neighbor discovery in asynchronous sensor networks [35] are other approaches found in the literature.

4. Summary of Important Techniques Found in Literature

This section provides some of the essential approaches found in the literature. It provides the techniques used by researchers, their advantages, limitations, and the simulation environment used by them.

As presented in Table 1, many neighbor discovery protocols are summarized. From the review of these techniques, the following are the research gaps identified.

5. Performance Comparison among different NDP

This section provides a comparative study of different NDP with respect to energy efficiency.

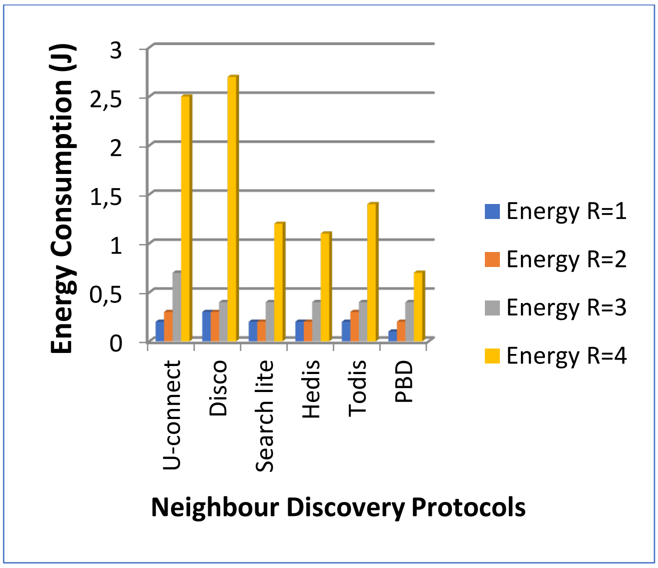

Figure 5. Comparison among different NDP

As shown in Figure 5, the graph shows a comparison of the average energy consumption of different scenarios. The x-axis shows that asymmetric ratio with different values. Y-axis shows that energy consumption. In this graph, DISCO has the highest energy consumption at R=10. PBD has the lowest energy consumption at R=10. Energy consumption at R=5 high for U-connect and low for PBD.TODIS has the highest energy consumption at R=2. Searchlight has the lowest energy consumption at R=10. Energy consumption at R=1 high for DISCO and low for PBD.

This section provides a summary of findings with respect to the recent state of the art on continuous neighbour discovery in WSN. The schemes or protocols summarised here include U-Connect, DISCO, SearchLight, Hedis, Todis, and PBD. Each protocol has its advantages and limitations, as provided in Table 1. Figure 5 presented in this section throw light into significant drawbacks of each neighbour discovery scheme. In other words, the research gap associated with each scheme is provided.

6. Discussion and Research Gap

Ref	Techniques	Advantages	Limitations	Simulation Tool
[36]	Optimal block design for asynchronous wake-up schedules	Scalability, energy efficiency and higher PDR	Need to improve worst –case latency and limited to specific duty cycles.	NS2
[37]	Review of schedule based asynchronous duty cycle mechanisms	Different metrics and trade-offs between power and latency are known.	Node residual energy is not considered for designing duty cycle.	TOSSIM
[38]	Code-based approach to ND	Improved latency	Worst-case latency is still high	Simulation study
[39]	Block design based ND	Enhanced latency performance and energy efficiency.	Node remaining energy is not considered for new block design.	TOSSIM
[40]	Nested Block Design based ND for WSN	Low latency	Need improve the efficiency of new block design.	Statistical simulations with R tool
[41]	Integer and non-integer schedules for duty cycles	Energy efficiency and reduced latency	Needs to improve the worst-case latency	Simulation study
[7]	Cod-based ND approach	Improved worst-case latency	Need to improve worst –case latency and limited to specific duty cycles.	Simulation study
[8]	DISCO protocol	Improved energy efficiency and latency	Worst-case latency is still high	Simulation study
[9]	U-Connect protocol	Guarantees a common active slot, energy efficiency, improved latency	Worst-case latency is still high	Simulation study
[42]	SearchLight	Matrix-based solution, simple for implementation	Compromises energy efficiency and latency	Simulation study
[29]	Hedis and Todis protocols	Low error rate	It needs more slots, compromises energy efficiency and latency.	Simulation study
[33]	Prime Block Diagram	Improved energy efficiency and reduced worst-case latency.	Not useful in route discovery when BIBD blocks are not available for specific duty cycles. It has a limitation in generating discovery schedules for a wide range of duty cycles.	Simulation study with TOSSIM

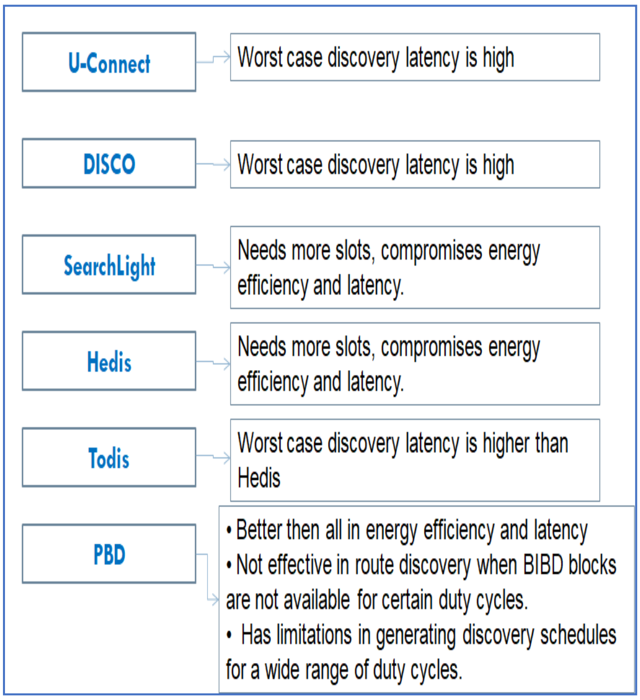

Figure 6. Summary of findings

As presented in Figure 6, it is evident that the existing neighbor discovery method is known as Prime Block Design (PBD) [12], extends Balanced Incomplete Block Design (BIBD) based protocol [1]. The main problem with PBD is that it is not useful in route discovery when BIBD blocks are not available for specific duty cycles. Moreover, it has limitations in generating discovery schedules for a wide range of duty cycles. This is a challenging problem to be addressed. Our future work focuses on overcoming the drawbacks of the PBD method used for neighbour discovery.

7. Conclusion and Future Work

Wireless Sensor Network intended to have long term monitoring applications should have an efficient neighbor discovery protocol. During deployment of sensor networks, it is an essential and challenging task to discover neighbors. The neighbor discovery is not a one-time event in the wireless sensor network (WSN) applications since a new batch of sensors can be deployed at any time during the mission. Thus, the latency and energy efficiency of maintaining neighbor nodes are directly related to the lifetime of sensor nodes with tiny batteries, and the main design issue of neighbor discovery protocols (NDPs) has been reducing discovery latency without sacrificing energy efficiency. This paper has made a review of different ND techniques. The existing neighbor discovery method known as Prime Block Design (PBD) [57] extends Balanced Incomplete Block Design (BIBD) based protocol. The main problem with PBD is that it is not useful in route discovery when BIBD blocks are not available for certain duty cycles. Moreover, it has limitations in generating discovery schedules for a wide range of duty cycles. This is a challenging problem to be addressed by considering it for our future work.

References

[1] T. Meng, F. Wu, G. Chen, On Designing Neighbor Discovery Protocols: a Code-based Approach, in: Proceedings INFOCOM, 2014.

[2] Wei Sun, Zheng Yang, Xinglin Zhang, and Yunhao Liu. Energy-Efficient Neighbor Discovery in Mobile Ad Hoc and Wireless Sensor Networks A Survey. IEEE Communications Surveys & Tutorials. 2014, 16 (3), p1-12.

[3] Sangil Choi, Wooksik Lee Teukseob Song, and Jong-Hoon Youn. (2015). Block Design-Based Asynchronous Neighbor Discovery Protocol for Wireless Sensor Networks. *Journal Of Sensors*. 2015, p1-13.

[4] Ying Qiu, Shining Li, Xiangsen Xu, and Zhigang Li. Talk More, Listen Less: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks, 2016, p1-9.

[5] Sixia Chen, Alexander Russell, Ruofan Jin, Sudarshan VasudevanYanyuan Qin, Bing Wang , Asynchronous Neighbor Discovery on Duty-cycled Mobile Devices: Integer and Non-Integer Schedules. *ACM*, 2015, p1-10.

[6] Athar Ali Khan, Mubashir Husain Rehmani, and Yasir Saleem. Neighbor Discovery in Traditional Wireless Networks and Cognitive Radio Networks: Basics, taxonomy, challenges, and future research directions. Elsevier, 2015, p1-18.

[7] Tong Meng, Fan Wu, Guihai Chen. Code-Based Neighbor Discovery Protocols in Mobile Wireless Networks. *IEEE*. 2016, 24 (2), p806-819.

[8] P. Dutta, D. Culler, Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications, in: Proceedings Sensys, 2008.

[9] Arvind Kandhalu, Karthik Lakshmanan, and Ragunathan (Raj) Rajkumar. U-Connect: A Low-Latency Energy-Efficient Asynchronous Neighbor Discovery Protocol. 2010, p1-12.

[10] Wei Sun, Zheng Yang, Keyu Wang, and Yunhao Liu. Hello: A Generic Flexible Protocol for Neighbor Discovery. IEEE Conference on Computer Communications, 2014, p540-548.

[11] Lin Chena and Kaigui Bianb. Neighbor Discovery in Mobile Sensing Applications: A Comprehensive Survey. Elsevier, 2016, 31, p1-15.

[12] Tao Shang. A Comparison Between Neighbour Discovery Protocols in Low Duty-Cycled Wireless Sensor Networks. Journal Of Computer science and mobile computing; 2015, 4 (2), p265-271.

[13] Xiangfa Guo, Binbin Chen, and Mun Choon Chan. Analysis and Design of Low-Duty Protocol for Smartphone Neighbor Discovery, 2017; p1-15.

[14] Riccardo Pozza, Michele Nati, Stylianos Georgoulas, Klaus Moessner, And Alexander Gluhak. Neighbor discovery for opportunistic Networking in the Internet of Things Scenarios: A Survey. IEEE. 2015, 3 (.), p1101-1131.

[15] Chew, Kelvin T. Symmetric link key management for secure neighbor discovery in a decentralized wireless sensor network, 2017, p1-104.

[16] Hao Ling And Siqian Yang. Anole: An Adaptive Neighbor Discovery Under Urban Environments. IEEE. 2018, 6, p64817-64827.

[17] Hao Cai and Tilman Wolf. Self-Adapting Quorum-Based Neighbor Discovery in Wireless Sensor Networks. IEEE Conference on Computer Communications, 2018, p324-332.

[18] Liangxiong Wei, Weijie Sun, Haixiang Chen, Ping Yuan, Feng Yin, Qian Luo, Yanru Chen, and Liangyin Chen. A Fast Neighbor Discovery Algorithm in WSNs†. Sensors, 2018, p1-18.

[19] Marjan Radi, Behnam Dezfouli, Kamalrulnizam Abu Bakar, and Shukor Abd Razak. (2014). Integration and Analysis of Neighbor Discovery and Link Quality Estimation in Wireless Sensor Networks. Hindawi Publishing Corporatione Scientific World Journal,2014, p1-24.

[20] Lin Chena and Kaigui Bianb. Neighbor discovery in mobile sensing applications: A comprehensive survey. Elsevier . 2016, 31 , p1-15.

[21] Georgios Z. Papadopoulos,b, Vasileios Kotsioua,d, Antoine Gallaisa, Periklis Chatzimisiosc, and Thomas Noel. Low-Power Neighbor Discovery for Mobility-Aware Wireless Sensor Networks. Elsevier. 2016, p1-32.

[22] Behnam Dezfouli, Marjan Radi, Shukor Abd Razak, Kamin Whitehouse, Kamalrulnizam Abu Bakar, and Tan Hwee-Pink. Improving broadcast reliability for neighbor discovery, link estimation, and collection tree construction in wireless sensor networks. Elsevier. 2014, 62, p101-121.

[23] Luhutyit Peter Damuut. X-Raying Neighbour Discovery In A Wireless Sensor Network Comprising Of Non-Homogeneous Nodes. ScienceWorld Journal. 2017, 12, p41-46.

[24] Tomás Robles, Borja Bordel, Ramón Alcarria, and Diego Mart ín. Mobile Wireless Sensor Networks: Modeling and Analysis of Three-Dimensional Scenarios and Neighbor Discovery in Mobile Data Collection, 2017, 35, p1-38.

[25] R.Saravanan, Dr. C. Parthasarathy, and Mallupalli Sathish. Secure and Efficient Data Search using Continuous Neighbor Discovery Algorithm in MANET. 2015, 5, p117-120.

[26] Fan Wua, Guobao Suna, and Guihai Chen. On Multi - packet Reception Based Neighbor Discovery in low- Duty-cycle Sireless Sensor Networks, 2015, p1-10.

[27] L. Chen, K. Bian, M. Zheng. Heterogeneous Multi-channel Neighbor Discovery for Mobile Sensing Applications: Theoretical Foundation and Protocol Design, in: Proceedings Mobi Hoc, 2014.

[28] Lin Chen, Ruolin Fan, Kaigui Bian, Lin Chen, Mario Gerla, Tao Wang, and Xiaoming Li, On Heterogeneous Neighbor Discovery in Wireless Sensor Networks, in: Proceedings INFOCOM, 2015.

[29] L. Chen, R. Fan, K. Bian, L. Chen, M. Gerla, T. Wang, and X. Li, "A Heterogeneous Neighbor Discovery in Wireless Sensor Networks," in Proc. IEEE INFOCOM, 2015, pp. 693-701.

[30] R. Pozza, M. Nati, S. Georgoulas, K. Moessner, A. Gluhak. Neighbor Discovery for Opportunistic Networking in Internet of Things Scenarios: a Survey, IEEE Access 3, 2015, 1101–1131.

[31] A. Hess, E. Hyytia, and J. Ott, " Efficient Neighbor Discovery in Opportunistic Mobile Networking using Mobility Awareness," in Proc. 6th Int. Conf. Commun. Syst. Netw. (COMSNETS) 2014, p1–8.

[32] Wei Sun, Zheng Yang, Xinglin Zhang, and Yunhao Liu. Energy-Efficient Neighbor Discovery in Mobile Ad Hoc and Wireless Sensor Networks: A Survey. IEEE Communications Surveys &Amp; Tutorials, 2014, 16, p1448-1459.

[33] Woosik Lee, Ik-Hyun Youn, Teukseob Song, Namgi Kim, and Jong-Hoon Youn. Prime Block Design for Asynchronous Wake-Up Schedules in Wireless Sensor Networks. *IEEE*. 2014, p1-4.

[34] Liangyin Chen, Yuanchao Shu, Yu Gu, Shuo Guo, Tian He, Fan Zhang, and Jiming Chen. Group-based Neighbor Discovery in Low-duty-cycle Mobile Sensor Networks. *IEEE Transactions on Mobile Computing*. 2015, p1-14.

[35] Reuven cohen and Boris kapchits. “Continuous Neighbour Discovery in Asynchronous Sensor Networks” IEEE Transactions on Networking. 2011, 19, p1-11.

[36] R. Zheng, J.C. Hou, and L. Sha, “Optimal Block Design for Asynchronous Wake-Up Schedules and its Applications in Multi-hop Wireless Networks,” IEEE Trans. Mobile Comput., 2006, vol. 5, no. 9, pp. 1228-1241.

[37] R.C. Carrano, D. Passos, L.C.S. Magalhaes, and C.V.N. Albuquerque. Comprehensive Analysis on the use of Schedule-Based Asynchronous Duty Cycling in Wireless Sensor Networks, Ad Hoc Networks, vol. 2014, 16, pp. 142-164.

[38] T. Meng, F. Wu, and G. Chen, “Designing neighbor discovery protocols: a code-based approach,” in Proc. IEEE INFOCOM, 2014, pp. 1689-1697.

[39] S. Choi, W. Lee, T. Song, and J. Youn, “Block Design-Based Asynchronous Neighbor Discovery Protocol For Wireless Sensor Networks,” Journal of Sensors, vol. 2015, Article ID 951652, 2015.

[40] R.C. Carrano, D. Passos, L.C.S. Magalhaes, and C.V.N. Albuquerque, “Nested Block Designs: Flexible and Efficient Schedule-Based Asynchronous Duty Cycling,” Computer Networks, 2013, vol. 57, no. 17, pp. 3316-3326.

[41] S. Chen, A. Russell, R. Jin, Y. Qin, B. Wang, and S. Vasudevan, “Asynchronous Neighbor Discovery on Duty-Cycles Mobile Devices: Integer And Non-Integer Schedules,” in Proc. ACM MobiHoc, 2015, pp. 47-56.

[42] M. Bakht, M. Trower, and R. Kravets, “Searchlight: won’t you be my Neighbor,” in Proc. ACM MobiCom, 2012, pp. 185-196.