DESIGN A SLOT-SCHEDULED BROADCAST PROTOCOL FOR WSNs TO DELIVER A MESSAGE IN RELIABLE AND ENERGY-EFFICIENT MANNER

THIẾT KẾ GIAO THỨC TRUYỀN BẢN TIN ĐIỀU KHIỂN HIỆU QUẢ NĂNG LƯỢNG VÀ TIN CẬY CHO MẠNG CẢM BIẾN KHÔNG DÂY

Ta Van Khoe[1,*], Tran Huu Phi[1]

DOI: https://doi.org/10.57001/huih5804.2023.009

ABSTRACT

A command delivery is a common operation and plays an essential role in wireless sensor networks. This can be simply achieved if every node that receives a command for the first time rebroadcasts the received command. However, it incurs high energy consumption and unpredictable end-to-end message delay since nodes must remain active. A slotted broadcast approach tackles these shortcomings by using a distinct time slot to every node but decreases the reliability and does not respond effectively to the dynamic networks. Our approach combines the advantages of the two approaches where one sharable slot is allocated to each tree level for competing and an on-the-fly mini-slot scheduling algorithm for scheduling mini-slots to nodes at the same level is devised to further reduce the possibility of message collision. Extensive simulation results proved that the approach improves the reliability of message delivery and conserves the energy significantly, even against the change of topology.

Keywords: *Slotted broadcast, spanning tree, node mobility, energy efficiency.*

TÓM TẮT

Với mạng cảm biến không dây (WSNs), việc truyền các bản tin điều khiển là hoạt động phổ biến và đóng vai trò rất quan trọng. Điều này có thể đạt được nếu mỗi nút truyền lại bản tin điều khiển đã nhận. Tuy nhiên, giải pháp này sẽ trả giá bằng năng lượng tiêu hao và trễ truyền bản tin lớn vì các nút phải luôn duy trì trạng thái thức. Việc phân khe thời gian truyền cho từng nút có thể tránh được các vấn đề này tuy nhiên nó sẽ giảm độ tin cậy của bản tin truyền đặc biệt khi có sự di chuyển của các nút cảm biến. Giải pháp của chúng tôi là kết hợp ưu điểm của hai phương pháp trên bằng cách chỉ định một khe thời gian dùng chung cho các nút có cùng số bước tới nút chủ để cạnh tranh cho quá trình truyền và đề xuất một thuật toán để chỉ định riêng một khe thời gian nhỏ cho mỗi nút để giảm xung đột. Các kết quả mô phỏng chứng minh rằng giải pháp đưa ra có thể nâng cao độ tin cậy của bản tin cần truyền và giảm được sự tiêu thụ năng lượng, và có thể ứng dụng trong WSNs động.

Từ khóa: *Truyền theo khe thời gian, cây truyền, sự di chuyển các nút, hiệu quả năng lượng.*

1. INTRODUCTION

In monitoring applications using wireless sensor networks (WSNs), a server is required to reliably broadcast a command to node(s) with time bound. However, this is not easy because the condition of links in WSNs is time-varying due to node mobility, internal interference, and external interference by other networks [1]. Energy efficiency is another critical issue since the replacement of batteries is often time-consuming and inconvenient.

To date, many protocols have been put out to guarantee the reliability of message delivery. A flooding scheme is one brute-force approach for reliable message delivery, in which a node retransmits the first received message. Glossy method [2] exploits the notion of the constructive interference, make concurrent transmissions of the same packet at the same time. In these schemes, a node may receive the duplicate messages from different neighbors, result in higher the consumption of energy and longer broadcast latency. Therefore, a duty cycle working method [3] has been proposed to reduce the energy consumption by alternating sleep and active mode. However, the solutions require each node reports the maximum delivery delay of its downstream nodes to the sink. Then the sink triggers the tree adjustment procedure so that every node searches for the best parent among its neighbors that considers an energy cost.

In the location-based method, one [4] relies on the construction of a connected dominating set (CDS) such that the flooding is done if a node in the CDS. Another [5] employs CDS and makes a slot scheduling to reduce collision. However, construction an optimal CDS is NP-complete [6]. Whereas the goal of paper [7] is to assign distinct slots to the nodes in CDS so that every node rebroadcasts a message without causing collision. Instead of finding CDS beforehand, they try to assign and adjust a slot number to a node depending on the distance from the node to a sink to resolve the collision. This approach incurs high overhead to build two-hop information for collision

---

[1] Le Quy Don Technical University

[*] Email: tavankhoe@gmail.com

Received: 05/01/2023
Revised: 15/02/2023
Accepted: 24/02/2023

avoidance and can cause the waste of slots. In the slotted-based approach, the authors in [8] proposed reliable slotted broadcast protocol (RSBP) in which a distinct slot is assigned to each node except for leaf nodes such that the lower tree level a node is at, the earlier broadcast slot it is allocated. This approach gives a predictable end-to-end delay of a message as well as an efficient energy management; however, it suffers from the reduced reliability since every node has only one chance of receiving a broadcast message.

This paper presents a slot-scheduled broadcast protocol (SSBP) that combines the strengths of the flooding and the sharable slot protocol [9] such that every node can generate its own sharable slot, making the sharable slot topology-independent. Moreover, we devise a new mini-slot scheduling algorithm to allocate mini-slots to nodes within a sharable slot in a fully distributed manner where a mini-slot is the slice of time enough to broadcast one message. Advantages of the protocol are several-fold. A node can still have multiple chances of receiving an identical message from different nodes at one level lower, thereby increasing the reliability of a message delivery. Furthermore, it can reduce energy consumption significantly by managing its duty cycle such that it only wakes up at the beginning of its own sharable slot and if a node rebroadcasts a message, it goes to sleep immediately. We can further reduce the possibility of message collision by allocating the mini-slots for rebroadcasting.

Simulation results shown that the proposed approach may achieve a high ratio of packet delivery and low energy consumption comparable to the flooding scheme, and the slotted approach, respectively, against changes in topology and different degrees of interference.

2. NETWORK MODEL

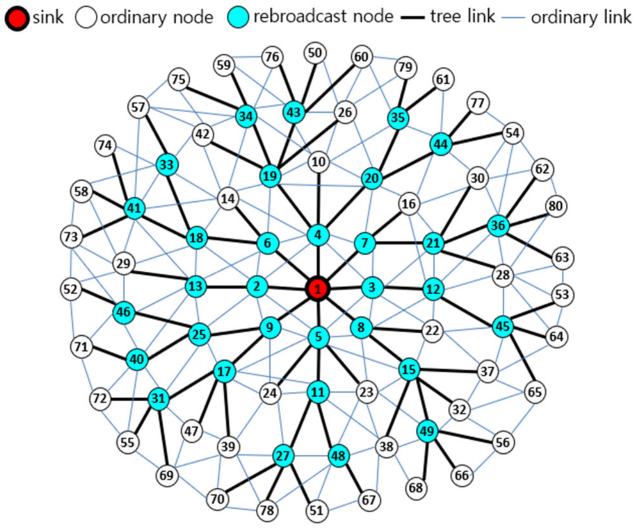


Figure 1. Network model with tree topology

A WSN consists of a number of sensor nodes and a sink. The sink is wall-powered whereas every sensor node is battery-powered and has a constrained communication range for energy saving. From the sink, the nodes can build a multi-hop tree. A tree-node is one that belongs to a tree; otherwise, it is an orphan-node. A link exists between two nodes that can communicate directly with each other, and it can be disrupted due to node failure, node mobility, battery depletion, or the intervention of some obstacles. A link between a parent and its child is called a tree-link.

Fig. 1 shows an example network with tree topology where a sink is located in the middle of the network and nodes are distributed around the sink. This paper aims at designing a broadcast protocol in which a sink sends a message in an energy-efficient and reliable manner to all nodes with delay bound. In fact, if the broadcast nodes rebroadcast a command message, all nodes can receive the message; however, node 31 can fail to receive a message if nodes 17 and 25 transmit the message simultaneously due to collision.

3. THE SLOT-SCHEDULED BROADCAST PROTOCOL (SSBP)

3.1. Protocol structure and tree structure

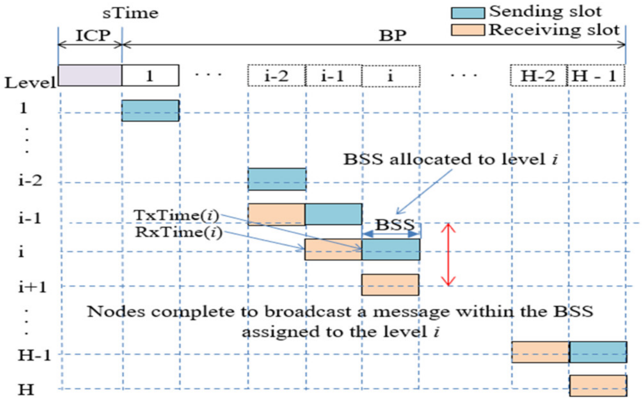


Figure 2. Protocol structure and the allocation of a broadcast sharable slot

As shown in Fig. 2 [9], the protocol begins with the initial construction period (ICP) and continues with the broadcast period (BP). Time synchronization, tree establishment, and slot scheduling are conducted sequentially during ICP. BP begins with a globally synchronized time, sTime. During BP, the broadcasting is performed, beginning with a sink at the lowest level and proceeding to the nodes at the next level progressively. With a tree of depth H, BP is divided into (H-1) broadcast sharable slots (BSS). During the i^{th} sharable slot allocated to level i, the node(s) at level i are in the broadcasting mode and each node goes to sleep immediately after rebroadcasting while those at level i+1 are in the receiving mode and each node goes to sleep immediately after receiving.

A tree is constructed by using control messages. Initiating, a sink broadcasts a tree construction request message (TCR). When a node receives the TCR, it attempts to join the sink by sending a join request message (J-REQ). When the sink receives a J-REQ, it adopts the node as a child and responds with an acknowledgement message (ACK). Upon receiving the ACK, a node is a tree-node, and its parent is the sink. Similarly, an orphan-node that overhears the J-REQ can join the tree, through the two-way handshaking process of J-REQ and ACK. This join operation continues until no orphan-node that overhears the J-REQ.

3.2. Sharable Slot scheduling

In Fig. 2, the sharable slots are of identical size and are shared for nodes at the same level. The receiving BSS assigned to nodes at level i must overlap with the sending BSS assigned to nodes at level i-1, because nodes must receive a broadcast message while their parents transmit the message. Accordingly, a receiving time RxTime(x), a sending time TxTime(x), and a sleep time SleepTime(x) of node x are given as follows:

$$\begin{aligned} & R_x\,\text{Time}(x) = \text{sTime} + (l_x - 2)\times \text{length(BSS)},\ 2 \le l_x \le H \\ & T_x\text{Time}(x) = \text{sTime} + (l_x - 1)\times \text{length(BSS)},\ 1 \le l_x \le H-1 \\ & \text{SleepTime}(x) = \text{sTime} + l_x \times \text{length(BSS)} \end{aligned} \tag{1}$$

where length(BSS) denotes BSS size, and l_x denotes the level of node x. In fact, the time that node x goes to sleep is the completion time of broadcast that is not later than SleepTime(x).

3.3. Message broadcast

Once the BP starts, every node determines its own BSS in a fully distributed manner and remains sleep mode until it wakes up at its RxTime or TxTime. A sink, named s, at level 1 starts broadcasting a message at time TxTime(s) while its child, say x, at level 2 waits to receive the message at RxTime(x). Every node goes to sleep as soon as it receives the message and wakes up at its TxTime to broadcast the received message. The nodes at tree level i compete for broadcasting within the BSS [9] allocated to level i. Furthermore, we propose a mini-slot scheduling algorithm to distribute mini broadcast slots (mBSs) to the nodes. The mBS assignment is calculated on the fly. The approach can deal with the problem of message collision that multiple hidden nodes get an identical mBS.

3.3.1. Mini-slot scheduling

The slotted contention-based transmission approach aims at alleviating the possibility of collision by employing the slotted Aloha method. A BSS is divided into a number of mini broadcast slots (mBSs) as follows:

$$\text{BSS} = (\text{mBS}_1, \text{mBS}_2, ..., \text{mBS}_N) \tag{2}$$

where, a mBS is of a fixed size sufficient to broadcast one message and $N = \lfloor BSS / mBS \rfloor$. Nodes contend a channel for broadcasting at the beginning of mBS by using a delay function, rdelay. Every node waits for rdelay and starts transmitting if it senses that the channel is idle. Other while, a node is delayed to the next mBS for competition. This approach can still suffer from collision if the competition for broadcasting within the same mBS involves many neighboring nodes and/or any hidden node(s). Thus, it would be desirable to distribute the competition start times of the nodes over different mBS's. If a node generates a random number, it can calculate the start time of the chosen mBS (chosen_mSS) out of N mBS's corresponding to the random number by using the following formula:

$$\text{Chosen\_mBS}(x) = (\text{random}(1,\ N) - 1)\times \text{length(mBS)} \tag{3}$$

However, this way of slot allocation may result in some slots wasted without being allocated to any node. On the other hand, two or more hidden nodes may be assigned an identical mBS to incur the collision of a broadcast message at an identical receiver. Therefore, we propose a mini-slot scheduling (msSche) algorithm that cannot only reduce the waste of slots, but also reduce the probability that multiple hidden nodes get an identical mBS. The scheduling principle of the msSche is stated as follows

A node generates a slot schedule such that its children have different mBS's if possible.

A node piggybacks the slot scheduling information in its broadcast message and then broadcasts the message. Its children find its mBS from the slot schedule and other nodes just take only the message after ripping off the slot schedule. Thus, the contention among all the siblings becomes free only if the number of mBSs is not smaller than the number of children.

The msSche algorithm works as follows. Since a sink is the only one, it allocates mini-slots to its children in a balanced manner such that one half of the nodes are assigned slots from the start of the first half of the slots and another half of the nodes are assigned slots from the start of the second half. This is effective when the number of children is small. For example, if a sink has two children for 10 mini-slots, one gets the 1st mini-slot and another gets the 6th mini-slot. For a node that is assigned the ith slot, its children are assigned different mini-slots sequentially starting with the ith slot, thereby preventing the children (siblings) from getting the identical mini-slot. This way of scheduling slots can also reduce the waste of slots by distributing mini-slots as evenly as possible.

Suppose that the node i has n children expressed as C(i) = {i1, i2, …., in}. We apply algorithm 1 [9] to schedule the mini-slot for node i, Sche(i) as follows:

```
// k: the number of mBS's to schedule;  n: the number of children
// slotNumber(i) the mBS number of node i
If i is a sink then
    $\text{Sche(i)} = \{(\text{ij}, x) \mid x = j \bmod k,\ 1 \le j < \lfloor n/2 \rfloor\} \cup$
        $\{(\text{ij}, x) \mid x = (j - \lfloor n/2 \rfloor + \lfloor k/2 \rfloor) \bmod k,\ \lfloor n/2 \rfloor \le j \le n\}$
If i is an intermediate node then
    $\text{Sche(i)} = \{(\text{ij}, x) \mid x = (\text{slotNumber(i)} + j - 1) \bmod k,\ 1 \le j \le n\}$
```

Algorithm 1. The msSche algorithm to schedule min-slots to children

The mini-slot schedule by applying the *msSche* to the tree in Fig. 1 is summarized in Table 1. We can get the schedule as: Sche(1) = {(2, 1), (3, 2), (4, 3), (5, 4), (6, 3), (7, 4), (8, 1), (9, 2)}, Sche(2) = {(13, 1)}, Sche(3) = {(12, 2)}, Sche(4) = {(10,3), (19, 4), (20, 1)}, …, Sche(9) = {(17, 2), (25, 3)}, …, Sche(12) = {(45, 2)}, Sche(19) = {(26, 4),(34, 1),(42, 2),(43, 4)}, Sche(11) = {(27, 4), (48, 1)}. Note that, leaf nodes do not rebroadcast a message.

Table 1. mini-slot schedule obtained from the application of msSche to the tree in Fig. 1

mBS #	**1**	**2**	**3**	**4**
Nodes at level 2	2	3	4	5
	8	9	6	7
Nodes at level 3	13			
	15	~~22~~		
		12		
		17	25	
	20		~~10~~	19
			~~14~~	18
	~~23~~	~~24~~		11
	21			~~16~~
Nodes at level 4	~~29~~			
	~~32~~	~~37~~	~~38~~	49
	35	44		
	~~28~~	~~30~~	36	
		45		
		31	~~39~~	~~47~~
			40	46
	34	~~42~~	43	~~26~~
	41			33
	48			27
~~X~~: X does not rebroadcast a message				

3.3.2. Message broadcasting

When node i receives a message from its parent, it generates a mini-slot schedule Sche(i) based on its own mBS number on-the-fly. When it rebroadcasts the message within the scheduled mBS, it piggybacks Sche(i) in the message so that its children can know the schedule. On the other hand, a node may not receive a broadcast message from its parent during the BSS of its parent due to node mobility, link breakage, or message collision. Therefore, a node always saves the broadcast message that it overhears for the first time from other nodes. If it fails to receive a message from its parent during the BSS of its parent, it chooses one of mBS's randomly, generates a slot schedule for its children and broadcasts the previously saved message with the slot schedule using the selected mBS. In this way, the protocol works very effectively against the dynamic change of topology.

The algorithm provides a lot of parallelism in broadcasting. For example, among the nodes at level 3, nodes 11, 18, and 19 are assigned the same mBS and can rebroadcast the message simultaneously without causing collision at any receiver (see Fig. 1). Therefore, we can realize the reliable broadcast with the small number of mini-slots.

4. PROTOCOL EVALUATION

4.1. Simulation setup

The simulation was performed using QualNet version 5.0.2. The sink is stationary while sensor nodes are either stationary or mobile. The random waypoint mobility model with a speed of 1m/s and a pause time of 5 seconds is used for mobile node(s). The Ricean fading model with driving parameter K is used to evaluation the variation of link condition. Two simulation scenarios, S1 which is a square dimensional zone of $30\times 30\text{m}^2$, consists of 30 nodes and one sink and S2 which is a square dimensional zone of $100\times 100\text{m}^2$, consists of 75 nodes and one sink are considered. Transmission range of nodes in S1 and S2 are 10m and 28m, respectively. Payload size of a packet is 30 bytes.

For evaluation, SSBP is compared with a simple flooding approach (Flooding) and one slotted broadcast protocol [8] (RSBP) where Flooding shows the best reliability, and RSBP achieves the high optimization in energy consumption. The sink broadcasts a command message whose MAC payload is of 30 bytes within a broadcast period. We use three metrics for comparison: (1) Packet delivery ratio (PDR), (2) Packet processing load (PPL), and (3) Average energy consumption (AEC).

4.2. Determination of protocol values

According to the IEEE 802.15.4 physical layer standard [10], a node takes the following message transmission process: (1) set a random delay to avoid collision (rdelay) that corresponds to the sum of a physical layer processing delay (= 12 symbols) and the CCA checking delay (= 8 symbols); (2) transfer a message from the MCU to a radio chip buffer (t_{mr}); (3) turn the radio chip on (t_{on}); (4) check CCA (t_{CCA}); (5) set the physical layer processing delay (t_{ppd}); (6) transmit a message (t_{tx}); and (7) transfer a message from the radio chip to the MCU at the receiver size (t_{rm}). The size of mBS should be optimized to increase channel efficiency. Within mBS, each node goes through the above seven steps to broadcast a message successfully. Thus, the size of mBS is given as follow:

$$\text{Size(mBS)} \ \geq \ \text{rdelay} + t_{mr} + t_{on} + t_{CCA} + t_{ppd} + t_{tx} + t_{rm} \qquad (4)$$

where rdelay is given as follow:

$$\text{rdelay} = \text{random}(0,\ \text{CW}) \ \times \ \text{delay\_time} \qquad (5)$$

Based on the 802.15.4 physical layer standard with a transmission rate of 250kbps (= 0.032 ms per byte): delay-time = 20 symbols = 0.32 (ms), CW is a contention window.

A synchronization header, PHY header, MAC header and footer of 6 bytes, we can calculate t_{tx} for the MAC payload of p bytes as follows:

$$t_{tx} = 0.032 \ \times \ (6+p) \quad \text{(ms)} \qquad (6)$$

With the transmission rate of 250 Kbps, 1 symbol = 0.016ms, thus $t_{CCA} = \ 8\ \text{symbols} = \ 0.128\text{(ms)}$ and $t_{ppd} = \ 12\ \text{symbols} = \ 0.192\text{(ms)}$. Additionally, for the simulation environment $t_{mr}=0(ms), t_{on}=0(ms), \text{and}\ t_{rm}=0(ms)$, from Eq. (4) we obtain:

$$Size(mBS) \geq 0.032 \times Max(CW) + 0.32 + 0.032 \times (6+p) \quad (ms) \tag{7}$$

The lower bound of BSS can be expressed as follows:

$$Size(BBS) \geq Size(mBS) \times nCnodes \quad (ms) \tag{8}$$

where nCnodes is the number of competing node and it can be estimated using [9].

4.3. Simulation results

4.3.1. Variation of mini-slot number

We used scenario S1 to verify the number of mini-slots, N, in a sharable slot. The N value is proportional to the likelihood of a collision and delay. If we increase N, number of nodes that generate the same mini-slot will obviously be reduced, resulting in a higher possibility of successful transmission. However, the increase in N causes an increase in the size of BSS leading to a high end-to-end delay in message transmission. The SSBP is tested with a variation in the number of mini-slots in a non-fading environment.

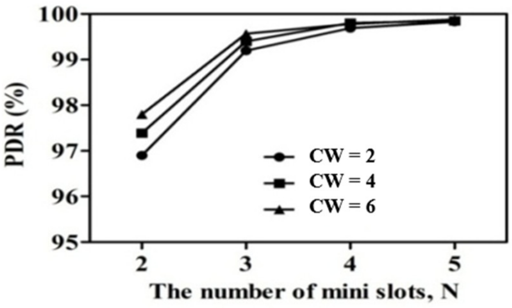
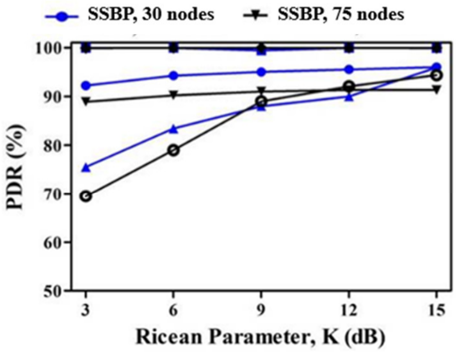


Figure 3. The likelihood of collision and number of mini-slots

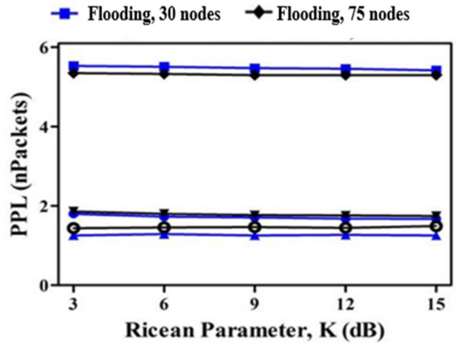
Referring to Fig. 3, the PDRs degrade sharply when N is less than 3. This implies that the lower bound of N is sound. On the other hand, an increase in N improves PDR, since it increases the number of chances to re-access the channel for transmission if the channel was busy in the previous mini slot. Furthermore, when the value of the contention window is increased with a fixed N, PDR also improved, since it relaxes the contention process in each mini slot. When N is greater than 4, PDR is almost saturated to about 99.8% at all cases of CW values. In consequence, the N of 4 is used for further performance evaluation.

4.3.2. Variation of driving parameter in fading effect model

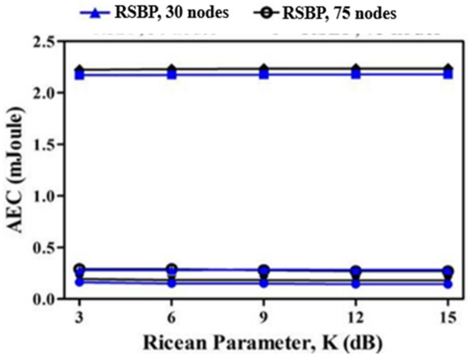
In Fig. 4, the performance of different protocols is given according to the increase in the degree of fading effect, K, from 3 to 15. The Fig. 4 (a) shows that the flooding is very reliable in any harsh environment. Meanwhile, SSBP also shows a good PDR even when K is extremely small (K = 3), slightly lower than the flooding protocol. The protocols allow a node to have a chance to receive the broadcast message from multiple nodes. On the contrary, it shows that the reliability of RSBT is not that reliable with the relatively high fading effect because the protocol relies on one small slot allocated to each broadcasting node.



(a) Packet delivery ratio



(b) Packet processing load



(c) Average energy consumption

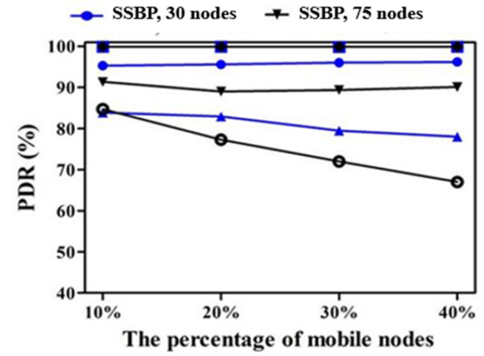
Figure 4. Performance comparison according to fading factor K

In Fig. 4 (b), it is shown that the PPL of the flooding method is substantially greater than that of the other two's, since a node may receive a message several times from surrounding nodes. With contrast, in SSBP, a node can go to sleep immediately upon received a message and finished broadcasting. It means that, every node receives and rebroadcasts only one message. In the larger scale network (scenario S2), the PPL of Flooding is very high, while the other two sustain very low PPL in both scenarios.

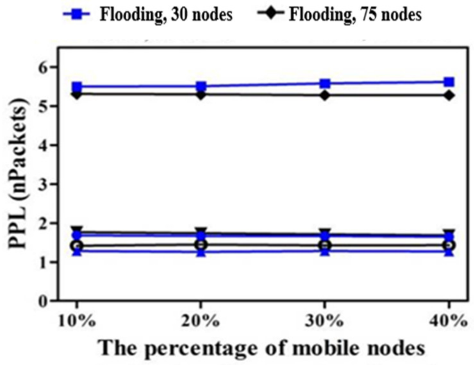
In Fig. 4 (c), SSBP shows the lower energy consumption considerably compared to Flooding, down to the slightly lower level to RSBP since the proposed protocol can manage energy consumption efficiently. Meanwhile, the Flooding consumes much higher energy since every node remains active all the time. The RSBP saves energy tightly such that a node and its children remain active only for the assigned slot and it uses an additional maintenance period during which all nodes stay active.

4.3.3. Variation of number of mobile nodes

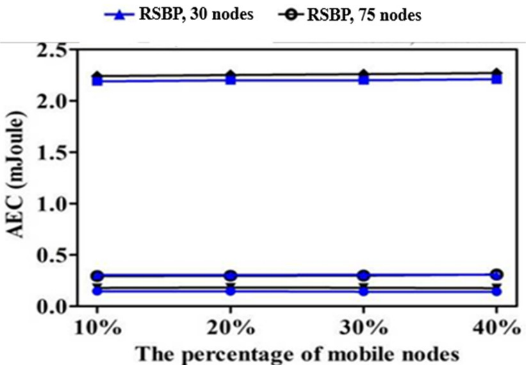
In order to examine the protocols under the change of topology, nodes are allowed to move arbitrarily by using the random waypoint model with a speed of 1mps and a pause time of 5s, and the K value is fixed at 10. The number of mobile nodes is from 10% to 40% of the nodes in the network. The simulation results are shown in Fig. 12. It is clearly seen that the PDR of Flooding and SSBP remain stable while that of RSBP decreases sharply with the increase in the number of mobile nodes. The high PDR of SSBP comes from the use of the sharable slot that lets nodes receive a message from multiple nodes. Furthermore, the SSBP allows a node to change levels easily and to respond quickly to node mobility. Otherwhile, in RSBP, if a node breaks a link with its parent, it cannot receive a broadcast message. Therefore, RSBP is not suitable for the dynamic change of topology.



(a) Packet delivery ratio



(b) Packet processing load



(c) Average energy consumption

Figure 5. Performance comparison according to the number of mobile nodes

5. CONCLUSION

We proposed a new reliable and energy-efficient broadcast protocol which employs a sharable slot and allocates mini-slots to nodes within a sharable slot on the fly. First, the protocol can manage energy consumption effectively by allowing a node to go into sleep mode. Second, it generates low control overhead since it uses mini-slots scheduling. Third, it can respond well to high level of interference and the change of topology because it uses a sharable slot. Intensive simulation results proved the soundness of the protocol.

REFERENCES

[1]. A. Sikora, V. F. Groza, 2005. *Coexistence of IEEE802.15.4 with other Systems in the 2.4 GHz-ISM-Band*. in 2005 IEEE Instrumentationand Measurement Technology Conference Proceedings, vol. 3, pp. 1786-1791, doi: 10.1109/IMTC.2005.1604479.

[2]. F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, 2011. *Efficient network flooding and time synchronization with Glossy*. in Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, pp. 73-84.

[3]. S. Wu, J. Niu, W. Chou, M. Guizani, 2016. *Delay-Aware Energy Optimization for Flooding in Duty-Cycled Wireless Sensor Networks*. IEEE Transactions on Wireless Communications, vol. 15, no. 12, pp. 8449-8462, doi: 0.1109/TWC.2016.2615296.

[4]. W. Peng-Jun, K. M. Alzoubi, O. Frieder, 2002. *Distributed construction of connected dominating set in wireless ad hoc networks*. in Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1597-1604 vol.3, doi: 10.1109/infcom.2002.1019411.

[5]. A. FathimaRamzi, N. Sabiyath Fatima, 2015. *Collision Optimized Broadcast Scheduling in Wireless Sensor Network*. International Journal of Computer Applications Volume 119, No.5.

[6]. O. Ugurlu, D. Tanir, E. Nuri, 2016. *A better heuristic for the minimum connected dominating set in ad hoc networks*. in 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), pp. 1-4, doi: 10.1109/icaict.2016.7991751.

[7]. D. Zhao, K.-W. Chin, R. Raad, 2014. *Minimizing broadcast latency and redundancy in asynchronous wireless sensor networks*. Wireless Networks, vol. 20, no. 3, pp. 345-360, doi: 10.1007/s11276-013-0607-8.

[8]. P. Van Vinh, H. Oh, 2012. *RSBP: A Reliable Slotted Broadcast Protocol in Wireless Sensor Networks*. Sensors (Basel, Switzerland), vol. 12, no. 11, pp. 14630-14646, doi: 10.3390/s121114630.

[9]. D.S. Yoo, V. K. Ta, B.T. Jang, H. Oh, 2019. *An Energy-Efficient Slotted Sense Multiple Access Broadcast Protocol for Reliable Command Delivery in Dynamic Wireless Sensor Networks*. Sensors, vol. 19, no. 5, doi: 10.3390/s19051236.

[10]. . Datasheet for the CC2420 radio component. Online: http://www.ti.com/product/CC2420.

THÔNG TIN TÁC GIẢ

Tạ Văn Khỏe, Trần Hữu Phi

Trường Đại học kỹ thuật Lê Quý Đôn