Dynamic Coordination of Energy and Hops in WSNs Using Reinforcement Learning Routing Algorithm

Jianyong Li[1], Huang Wei[2]

[1,2]Department of Computer and Information Science, Southwest University, Chongqing 400715, China

Keywords: Wireless sensor network; Routing algorithm; Reinforcement learning algorithm; Energy consumption

Abstract. In wireless sensor network, the existing reinforcement learning routing algorithm usually optimize single goal and the process of route establishment is complex. It also has problem of data forwarding control overhead. In this paper, we present a dynamic adaptive routing algorithm with feedback learning ability to balance the energy of wireless sensor network, to reduce the routing hops, and to reduce the establishment complexity. The algorithm will use the local routing information and the method of feedback to learn neighbors' state; routing reward values will be obtained by weighted calculation according to the energy information and the hop counts information; the optimal routing strategy will be obtained by updating the Q-value of routing table.

Introduction

Wireless sensor networks (WSNs) is a self-organized network which is composed of a large number of sensor nodes. In many fields such as military, industry and agriculture, it has a very broad prospects for development[1, 2]. Because WSNs is often limited by the energy supply and communication ability, so the design of routing algorithm is often challenging. Efficient usage of energy and reduction of hop counts of data transmission are important goals in WSNs routing algorithm design

Common WSNs routing protocols can be classified into four categories. Representative of the first kind is Flooding[3], it's forwarding rules is simple and implementation is easy. Flooding can reduce consumption of computational resources without the network topology information and complex routing discovery algorithm. But there are problems such that information explosion. A typical example of the second category is hierarchical routing which is a low energy adaptive clustering routing (LEACH)[4]. It need to control network topology with election of cluster nodes and it is responsible for the data fusion to reduce data traffic[5]. But cluster grouping brings extra overhead and coverage problem. The third class is data-centric routing which is based on the query, it's classic representative is DD (Directed Diffusion)[6] whose nodes and neighbor nodes communication without global topological information of each node, that reduce the amount of data traffic and energy consumption. But it's build of gradient is costly, and it's naming mechanism limits the range of application. The fourth class is based on the location information of routing, typical representative is GEAR[7]. It is combined with the geography information to forward data to the destination node according to the corresponding strategy which reduces routing overhead. But the node location need GPS which increase the cost.

Learning algorithm has distributed autonomous behavior and adaptability to environmental change, that making them very suitable for wireless sensor networks[8]. Literature [9] proposed a distributed reinforcement learning algorithm (DIRL) to enable sensor nodes to autonomous learning, to reduce energy consumption of finding routing path. Egorova Forster in [10] presents a learning algorithm based on feedback, the algorithm mainly use the feedback information to collect neighbor nodes routing information at the time of sink node statement, it make data reach more sink node effectively and avoid to send more data copy. But this method may get a longer path of data transmission. In order to maintain the network node energy balance, literature [11] proposed a routing algorithm based on reinforcement learning (RL), it adjust data transmission path dynamically to avoid a single node energy depletion, that make whole network energy equilibrium,

prolong the network lifetime.

Combining with Q-learning algorithm[12], this paper proposed a routing strategy based on feedback learning, a kind of Energy Consumption Balance and hops Less Adaptation Routing Algorithm (ECBHLA). It make WSNs energy equilibrium and reduce the hops of data transmission that will improve the overall network performance and enhanced the robustness of the network. The algorithm has very low overhead of routing and it is completely distributed.

Routing Algorithm With Feedback Based On Reinforcement Learning

Common WSNs routing algorithm is used to collect data from a source node, and then spread to a sink node. In the process of data transmission, one neighbor node can be chosen to forward data. The choices of neighbor nodes determine the routing path. Fig. 1 show a routing path from the source node to the sink node. Reinforcement learning algorithm for WSNs routing path finding process is similar to the markov decision process (MDP), MDP include:

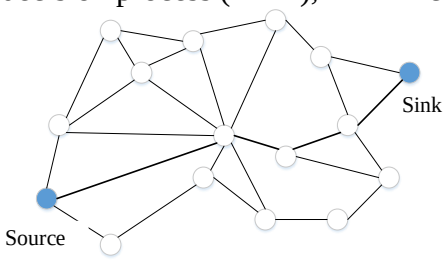

Fig. 1: Single source and sink network topology

(1) S, the state set

(2) A, the action set

(3) P: $S \times S \times A \rightarrow [0,1]$, $P(s' \mid s, a)$:

$\forall s \in S \quad \forall a \in A \quad \sum s' \in S \; P(s' \mid s, a) = 1.$

P (s' | s, a) represents the corresponding action probability value from s to s' state a.

(4) R:$S \times A \times S \rightarrow R$, R(s, a, s'), give a Movement value of reward.

In WSNs environment, each node can be regarded as a state *s* and neighbor state *s'*. From state *s* to *s'* has a corresponding action *a*, perform an action *a* means from *s* send data to *s'*. In the routing table, each item corresponding to a Q-value. $Q(s,a_1)$ on behalf the Q-value of the current node *s* routing to the next hop node *s'* to perform an action a_1. When forward data, traverse the Q-value in the routing table to choose a neighbor nodes, as shown in figure 2. Therefore, the main task of the learning strategy is to find a particular sequence of Q value.

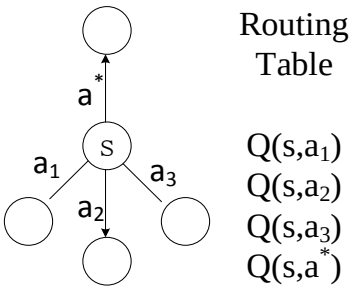

Fig. 2: state - action mapping

Use MDP quad (S, A, P, R) to establish a model agent. To solve the MDP is to find an optimal strategy:

$$\pi : S \rightarrow A$$

π on behalf of a sequence of mapping of a group of state to the action. Enable the sequence to achieve maximum value of the total reward equivalent to find a fixed set of Q-value.

Each time to perform action *a*, the agent will receive a “current reward point” and a “long-term reward value” which used to update the Q-value. The updated Q-value will affect the future choices

of routing. Update the Q-value using formula[13]:

$$Q(s,a) := (1-\alpha)Q(s,a) + \alpha\left[r(s,a) + \gamma \max Q(s',a')\right] \quad (1)$$

where α is the learning rate, it's value range in $0 < \alpha < 1$. It controls the updated Q-value of speed. γ , a factor to control the "long-term reward" value will contribution to Q-value. *r*(*s, a*) calculation is as follows:

$$r(s,a) = \sum_{s'} R(s,a,s') \quad (2)$$

The agent will try to move from one state to another state, until reach the target state. We call this is a discovery process. Agent keep record of quad (s, a, r, s') sequence which will learning the "experience". (s, a, r, s) means performed the action *a* by state *s* reach *s'* getting the reward value *r*. By the "experience", use the formula (1) update the Q-value. Agent perform actions according to Q-value, eventually get π:

$$s_0, a_0, s_1, a_1, s_2, a_2, s_3, a_3, s_4...$$

s_0 is an initial state, s_i is the middle state, a_i means that in state s_i can perform an action a_i

To receive data from other nodes in a network, sink node must broadcast request message (Request Message) to the network, declaring himself to be the sink node. This process is similar to DD's interest spread phase where each node forward the request message to the surrounding nodes after receives the request message. The content of the request message contains: *ID*, *SinkID*, *Hops*, and *Energy*. Fig. 3 shows the structure of a request message.

Fig. 3: Request message structure

ID indicate the node of forwarding the request message. *SinkID* item indicate the sink node, used to declare the sink node. *Hops* item represent the number of the request message is forwarded. *Energy* item labeled the nodes' remaining energy.

Each node maintains a routing table, as shown in Table 1. Each line of routing table represents a routing table entries which has four properties. The first line stores the ID of the sink node. The rest stores neighbor nodes. Energy properties represent the residual Energy of neighbor nodes, *Hops* represent minimum hops to forward data from neighbor nodes to the sink node. *Q-value* represent Q value of rule that choosing a neighbor node as the next-hop node to forward, Q-value obtained by Eq. (1).

Table 1: Routing table

	ID	Energy	Hops	Q-value
Sink	A	–	–	
Neighbor1	38	80	3	*
Neighbor2	46	64	5	*
⋮				
NeighborN	-	-	-	-

When a node receives a neighbor's request message, it will check whether the request message's ID in the neighbor's routing table or not. If yes, then update the neighbor items of *Energy* value and *Hops* according to the information in the request message. If not, add a neighbor item to the routing table and it's *ID* is the request message's ID, then use the information which come from request message to initialize the rest properties of the neighbor item. In Table [1], for example, the routing table is empty initially, after receiving request message from neighbor 38 and 46, add two table entries: *Neighbor1* and *Neighbor2*.

Source nodes send data to the neighbor nodes in flood way, neighbor nodes will send it's routing information back to the neighbor which forward data to it. The node who receives the feedback information need to calculate the reward values, *R (s, a, s')*. Considering the global Energy balance and real-time (hop less), *Energy* and *Hops* in the routing table design to *R* (*s, a, s '*) as parameters:

$$R(s,a,s') = \theta E_{s'} + (1-\theta)\frac{A}{H_{s'}} \quad (3)$$

θ is the weight parameter, it's value between 0 and 1. In order to prevent routing too biased towards the energy equilibrium and ignore the real time, or too biased towards real-time and ignore the energy equilibrium, θ suggested value is 0.6. *Es′* represent feedback energy values of neighbor node *S*. *Hs′* represent feedback hops values of neighbor node *S*. The term *A* is a constant value, *A* can be customized in the practical application. In the case of Table [1], when update the Q-value of Neighbor1 whose ID is 38, first calculate *R* (*s, a,* 46) and *R* (*s, a,* 38) according to the Eq. (3), then calculate the *Q*(*s, a*) according to Eq. (1) and Eq. (2).

In order to improve the energy utilization and reduce redundant transmission, we don't use the sink node periodically broadcast request data to drive the whole network nodes update their routing table. Here we use the method of notify sink node: when a node's energy content Eq. (4), the node forwards data with an update mark. Sink node sends the request message if received update mark.

$$\frac{E_0 - E}{E} \geq \rho \quad (4)$$

Where E_0 is the node's energy at the latest update moment, and *E* is the node's energy of current moment. The value of ρ can based on the dynamic variation of the network speed in the interval values between 10% ~ 20%.

Experiments

Network simulation tool is OMNeT++ 4.5, running in Linux operating system. The main simulation Settings: Node movement type is *StationaryMobility*, speed is zero. Initial energy of nodes is 1000mAh. Nodes' maximum send power, *pMax* is 0.15 mW. Fig. 4 is the topology of network which contents 50 nodes. The nodes (exclude sink node) every 30 seconds to produce a packet of 50 byte size in application layer. The network layer using the ECBHLA and Flooding. The physical layer using the IEEE 802.15.4 protocol. Table 2 lists the main configuration of each layer.

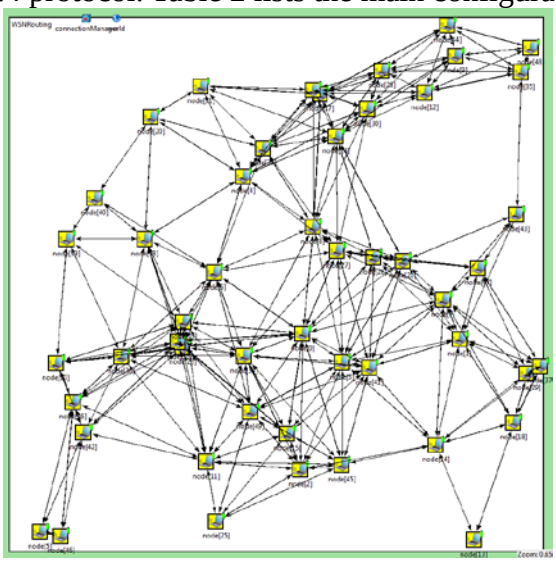

Fig. 4: Network topology

Table 2: The configuration

parameters	value
ApplicationType	SensorApplLayer
nicType	Nic802154_TI_CC2420
arpType	ArpHost

Because ECBHLA need to sink statement phase and learning phase, so the network energy consumption is larger than Flooding in the early. After establishing node's routing table, energy

consumption would be reduced. Figure 5 shows the energy dissipations of ECBHLA and Flooding respectively run 20 minutes, 30 minutes, 60 minutes and 90 minutes in 150 nodes network. Compared with Flooding routing protocol, ECBHLA in 0 ~ 40 minutes the consumption of energy is larger, With the increase of operation time, ECBHLA energy cost is relatively lower, compared with Flooding routing protocol, the overall energy consumption advantages are more obvious.

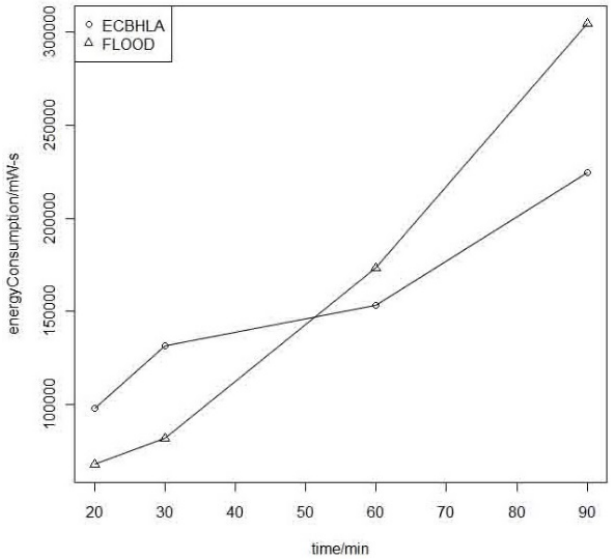

Fig. 5: Network energy consumption

Source nodes send data through multiple hops to the sink node, the sink node counts hops. Fig. 6 is the average hops statistics under the situation of containing 150 nodes topology network running 120 minutes. Due to data of Flooding protocol broadcast out simply, the path of the data reach the sink node is uncertain, so the average hops that data need to reach the sink node will not decrease along with the network running time. ECBHLA's routing behavior is similar to Flooding in the sink statement phase and learning phase, the average hop counts of data transmission is larger. But with the increase of network running time, the nodes' routing table gradually established, the average hop counts of ECBHLA will be reduced.

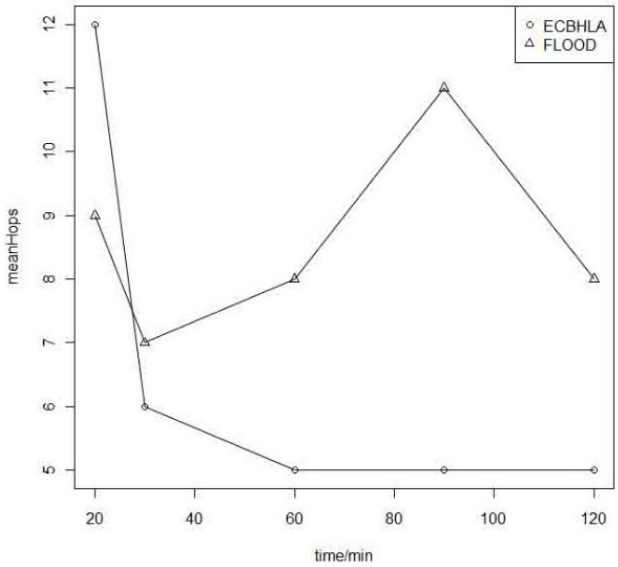

Fig. 6: Mean hops

Table 3: Data

	Energy consumption (mW-s)				Mean hops			
	20min	30min	60min	90min	20min	30min	60min	90min
ECBHLA	96530	136800	159800	210740	12	6	5	5
FLOOD	20047	55230	180937	291480	9	7	8	11

Table 3 is the statistics of the energy consumption and the average hops that ECBHLA and Flooding in the network which contains 150 nodes under the simulation of running 20 minutes, 30 minutes, 60 minutes and 90 minutes. From the point of data in the table, in the consumption of energy, with the increase of operation time, ECBHLA's growth of energy consumption is slow. 20 ~ 90 minutes, the growth rate of ECBHLA energy consumption is about 1400mW-s/min, and Flooding is about 3800mW-s/min. Given the average hop counts of data transmission, comparing ECBHLA and Flooding in the period of 30 ~ 90 minutes, the former is about 5.3, the latter is about 8.6. ECBHLA's fluctuation of the mean hops is more stable, the former variance is about 0.6 while the latter is about 6.3.

Conclusion

This paper analyzes the information processing mechanism of a single sensor in WSNs, proposes an energy equilibrium and less hops dynamic adaptive routing algorithm (ECBHLA) by combining with the enhanced learning algorithm of machine learning. ECBHLA use local routing information while it don’t need the whole topology information to learn neighbor state through the feedback and control the forward of data, nodes choose the neighbors which required less hops and has larger energy. ECNHLA’s energy consumption is larger at first, but it will decrease after the learning phase. In the network, data transmission delay can be measured according to the average hops and ECBHLA’s average hop counts remains relatively low in stable transmission phase.

Acknowledgements

Project supported by Chongqing integrated demonstration project (CSTC2013jcsf 10008) and Twelfth five year national support plan project: the key technology integration and demonstration of information service of rural Internet of things (No. 2012BAD35B08).

References

[1] Akyildiz,W Su, and Y Sankarasubramaniam. Wireless sensor networks: a survey[J]. Computer networks 38.4 (2002): 393-422.

[2] Jianzhong Li, Jinbao Li, Shengfei Shi. The concept of sensor networks and data management, problems and development [J]. Journal of software 14.10 (2003): 1717-1727.

[3] Marti, Mikls. Directed Flooding-routing framework for wireless sensor networks[C]. Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware. Springer-Verlag New York, Inc. 2004.

[4]Heinzelman, Wendi B, Anantha P, Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks[A]. Wireless Communications, IEEE Transactions on 1.4 (2002): 660-670.

[5] Limin Sun. Wireless Sensor Network[M]. Tsinghua university press co., LTD, 2005.

[6] Intanagonwiwat, Chalermek, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks[C]. Proceedings of the 6th annual international conference on Mobile computing and networking. ACM, 2000.

[7] Yu Yan, Ramesh Govindan, and Deborah Estrin. Geographical and Energy aware routing: A

recursive data dissemination protocol for wireless sensor networks[A]. Technical report ucla/csdtr-01-0023, UCLA Computer Science Department, 2001.

[8] Forster, Anna. Machine learning techniques applied to wireless ad-hoc networks: Guide and survey[C]. Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on. IEEE, 2007.

[9] Shah, Kunal, and Mohan Kumar. Distributed independent reinforcement learning (DIRL) approach to resource management in wireless sensor networks[C]. Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on. IEEE, 2007.

[10] Egorova-Forster A, Murphy A L. A feedback-enhanced learning approach for routing in WSN[C].Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference. VDE, 2007: 1-12.

[11] Forster A, Murphy A L. Balancing Energy expenditure in WSNs through reinforcement learning: a study[C].Proceedings of the 1st International Workshop on Energy in Wireless Sensor Networks (WEWSN), Santorini Island, Greece. 2008: 7pp.

[12] Watkins, Christopher JCH, and Peter Dayan. ”Q-learning[J].” Machine learning 8.3-4 (1992): 279-292.

[13] Wang, Ping, and Ting Wang. Adaptive routing for sensor networks using reinforcement learning[C]. Computer and Information Technology, 2006. CIT’06. The Sixth IEEE International Conference on IEEE, 2006.