ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 11, No. 3, 2016, *pp*.208-213

Optimal *k*-Anycast Routing Algorithm for Sleep-Wake Scheduling Wireless Sensor Networks

Xin Xu, Yun-li Gu, and Ya-ping Cheng
Jiangsu Engineering Center of Network Monitoring,
Nanjing University of Information Science & Technology, Nanjing 210044, China
(*Received April 10, 2016, accepted June 01, 2016,)*

Abstract. Finding optimal *k*-anycast routing paths for sleep-wake scheduling wireless sensor networks(WSN) is a NP-complete problem. Most previous research work only attends to optimizing the time delay during a hop or a single path. But in terms of the performance of the end-to-end delay, this scheme is not optimal. In this paper, an optimal *k*-anycast routing algorithm for sleep-wake scheduling WSN is proposed to solve the problem. In the proposed algorithm, base stations apply AODV-based multipath routing protocol to acquire *k*-anycast routing information, then genetic algorithm is applied to search the optimal *k* routing paths between nodes and bases stations. Since the proposed algorithm has the feature of global optimization ability, in contrast to previous algorithms, experiment results show that the proposed algorithm can reduce end-to-end delay more efficiently.

Keywords: *k*-anycast, genetic algorithm, wireless sensor networks, routing algorithm.

1. Introduction

In a large-scale WSN, if all sensor nodes send their monitoring data to only one base station (or sink node), the base station would become a bottleneck. A solution to this problem is to deploy multi base stations in a monitoring field. And for reliability, energy-consuming-balancing, load-balancing and security purpose, it is useful to ensure that each sensor node should send its monitoring information to any *k* of all base stations. This communication model is generally called *k*-anycast.

In the literature, only a few algorithms have been designed for *k*-anycast routing [1-7]. The work in [1-2] proposed a *k*-anycast model in a mobile Ad hoc network and described the implementation scheme. Then Wang [3] proposed a *k*-anycast communication model for wired networks, which maintains a *k*-anycast tree according to the "join" and "leave" requests of the member nodes. Dow [4] also established an anycast tree based on the clustering and virtual backbone to discover *k* services. The work in [5] proposed a distributed geographic *k*-anycast routing (GKAR) protocol for WSN, which can efficiently route data from a source sensor to any *k* destinations. Subsequently, Gao [6] proposed a *k*-anycast routing protocol for WSN based upon an anycast tree scheme. In the scheme, a source initiates to create a spanning tree reaching any one sink with source node as the root. In the work [7], *k*-anycast is applied in publish/subscribe-based information-centric network(PS-ICN).

Sleep-wake scheduling is an effective mechanism to prolong the lifetime of energy-constrained sensor networks. In this scheduling, sensor nodes periodically or aperiodically exchange information with their neighboring nodes. Such as Hsu[8] and Naveen[9] appropriately arrange sensor nodes to sleep when data transmission or reception is not needed. In sleeping state, the communication module of a sensor node is turned off; so the energy consumption is fairly low. Thus, sleep-wake scheduling is suitable for WSN.

But according to sleep-wake scheduling, nodes are not always under working state, and senders often have to wait the next stop node to wake, thus leading much sleeping delay. In a multi-hop WSN, sleeping delay in each hop will accumulate, thus ultimately leading a bad performance of end-to-end transmission delay. The objective of the study is to find *k* routing paths which have the optimal performance of end-to end delay.

2. Problem description

We model a WSN by a directed graph $G(V, E)$, where *V* is the set of *n* vertices, and *E* is the set of *m* edges. We denote *N* by sensor nodes number; *S* is the source node; $G(A)$ is *k*-anycast group (base stations) and $A_1, A_2, \cdots, A_M$ are *k*-anycast members; r_i represents the wake-up rate of node *i*. Wake-up rate means the number of rounds a node from start working to end sleeping in a unit time. In each sleep-wake cycle, each node should promise at least Δt working time. We represent $r = (r_1, r_2, \cdots, r_N)$ the wake-up rate vector of WSN. For energy balancing purpose, *k*-anycasts paths are disjoint with each other. That means, each node joins at most one *k*-anycast path to prevent the node's energy draining too fast.

k-anycast routing is a NP-complete problem. That means, we cannot find a finite polynomial solution. According the algorithm in [3], they would select the best path one by one (best path first scheme, BPF) until they find k paths. But in fact it is not the optimal solution. As shown in Fig. 1, assuming k=2, path $P_1 = S, n_1, n_2, n_3, A_1$ will be selected firstly according [3], then since node n_1 and node n_3 are used by path P_1, $P_2 = S, n_7, n_8, n_9, A_3$ will be selected as the second path (*k*-anycast paths must be disjoint paths). Obviously, the end-to-end delay performance of $P_3 = S, n_4, n_2, n_3, A_1$ and $P_4 = S, n_1, n_5, n_6, A_2$ is better than P_1 and P_2. Therefore, BPF scheme is not the best scheme.

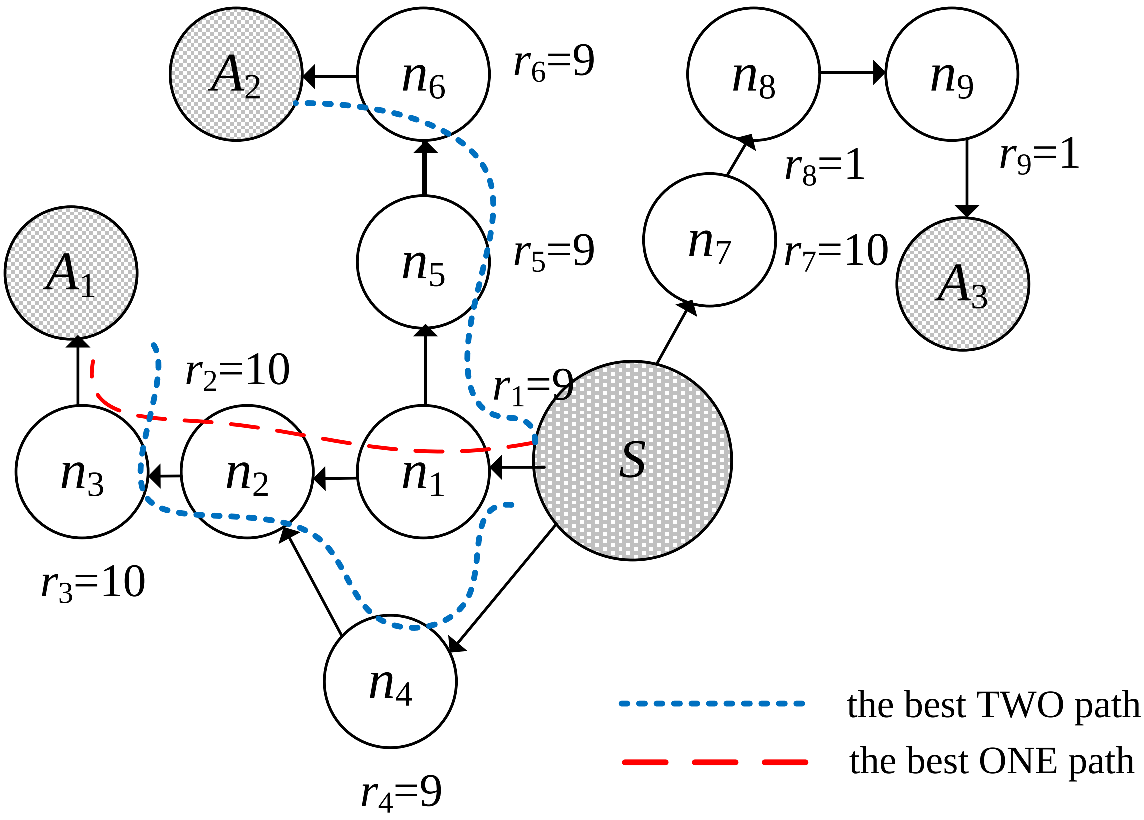

Fig 1. Illustration of *k*-anycast routing

Since *k*-anycast routing is a NP-complete problem, Kim [10-11] developed a delay-optimal anycast scheme. In the scheme, all nodes must follow a fixed sleep-wake scheduling and all source node must follow a fixed report task scheduling, and it turns the scheme into not practical (because the events such as nodes fall failure; select another path for energy balancing; clock drift; and so on). So in this paper, we propose an optimal *k*-anycast routing algorithm for sleep-wake scheduling WSN, and the proposed algorithm doesn't need the restriction on fixed report task scheduling.

3. Network model

For a *k*-anycast path $P = S, n_0, n_1, \cdots, n_L$, we have the end-to-end delay D_P as follows:

$$D_P = \sum_{i=0}^{L}(D_t(n_i) + D_s(n_i)) \qquad (1)$$

Where, S is the source node, $n_L \epsilon G(A)$, $D_t(n_i)$ represents the transmission delay at node n_i, and $D_s(n_i)$ represents the waiting delay at node n_i. As we know, while a node is in wakeup state, the waiting delay is 0; otherwise; while a node is in sleeping state, the average waiting delay is$(1 - r_i\Delta t)/2r_i$. Thus, we have the expected value of $D_s(n_i)$ as follows:

$$D_s(n_i) = (1 - r_i\Delta t)^2/2r_i \qquad (2)$$

While $r_i \in [0, 1/\Delta t]$, we know that $D_s(n_i)$ is monotone decreasing. And from Eqs.1, we also know that increasing the wake-up rate of a node can decrease the average waiting delay at that node.

Then we illustrate the end-to-end delay performance of k-anycast routing. Let P be the set of all paths to k-anycast members. Supposing the subset T (the paths are $\{P_1, P_2, \cdots, P_K\}$, and they are disjoint) is selected as k-anycast paths, the end-to-end delay performance $D(T)$ is defined as

$$D(T) = D(P_1, P_2, \cdots, P_K) = \max(D(P_1), D(P_2), \cdots, D(P_K)) \quad (3)$$

And the end-to-end delay performance $D(P)$ is defined as

$$D(P) = \min(\max(D(P_1), \cdots, D(P_K)), \cdots, \max(D(P_i), \cdots, D(P_{i+K-1}), \cdots) \quad (4)$$

Take Fig. 1 for example, k=2 and $P = \{P_1, P_2, P_3, P_4\}$. Since P_1 is joint with P_3 and P_4

$$D(P) = \min(\max(D(P_1), D(P_2)), \max(D(P_3), D(P_4), \max(D(P_2), D(P_3)), \max(D(P_2), D(P_4))) \quad (5)$$

Assuming Δt=0.02 unit time, then the expected value of end-to-end delay (omit the transmission delay) of k-anycast path P_1, P_2, P_3, P_4 are respectively 0.1014, 0.9924, 0.1014, 0.1121. So, we have $D(P)$ =0.1014, the best value comes from the paths $\{P_3, P_4\}$. In contrast, according BPF scheme [3], path P_1 will be selected firstly. And since P_1 is joint with P_3 and P_4, path P_2 will be selected as the second path. We know $D(P_1, P_2)$ =0.9924, so in fact BPF scheme is not the best scheme.

But finding the optimal k anycast paths is a NP-complete problem, so we cannot find a finite polynomial solution for this problem. Genetic algorithm is an effective technique for global optimization, and it is often applied to resolve complex optimal design problems. And only needing objective function and fitness function information, genetic algorithm can run. Thus, in the paper we apply genetic algorithm for searching the optimal route.

4. Optimization by genetic algorithm

In general, sensor nodes have less battery power and computing capability than base stations, so we let base stations take the jobs of the optimization calculation.

Our algorithm is designed as follows:

- Base stations obtain all k-anycast routing paths (from all sensor nodes to all base stations) information by AODV-based (Ad hoc On-demand Distance Vector Routing) multipath routing protocol;
- Each sensor node should broadcast its latest wake-up rate while its wake-up rate has been changed, and base stations obtain and keep the latest wake-up rate of all sensor nodes;
- Base stations calculate the optimal k-anycast routing paths (from each sensor node to base stations) by genetic algorithm, and report the result to all sensor nodes;
- While the energy power of a node is lower than a certain threshold value, the node should reduce its wake-up rate and report the information to all base stations before the node goes to the next sleep;
- Base stations need to re-calculate the optimal paths if there is a change about the wake-up rate of nodes.

The steps of optimization calculation by genetic algorithm are as below:

- **Initialization of chromosomes.** In the algorithm, any k paths from the source node to base stations are a feasible solution. Each node has a number and the nodes are used to encode a path as a string expressed by the order on numbers. A chromosome corresponds to a possible solution of the optimization problem. Since the chromosome is represented by a string bits, special crossover and mutation operators can be adopted. The initial population of chromosomes is randomly generated.
- **Evaluation function.** The choice of a fitness function is usually very specific to the problem under condition. The evaluation function of a chromosome measures the objective cost function. The cost of a path indicated by the chromosome is used to calculate its fitness. Since the fitness should increase as the cost decreases. Thus, the fitness function of a solution is evaluated as defined in Eqs.5.
- **Crossover.** Crossover recombines two parent paths to produce two new children paths in the next generation. Two points crossover is applied. Both parent paths are divided randomly into three parts respectively and recombined. The middle part of the first path between crossover bit positions and the middle part of the second path are exchanged to produce the new children.
- **Mutation.** The mutation process is also applied to flip randomly a bit position in the chromosome.

5. Experiments and analysis

In this section, we will compare the performance of the proposed algorithm with BPF(Best Path First) and SPF(Shortest Path First). In the simulation experiments, nodes and 8 sink nodes (base stations) are randomly deployed in a two-dimensional space. We set the transmission rate to be 640kb per unit time; pack size to be 512B. So, the time delay for transmission at each hop is 0.00625 unit time (for convenience, we omit the time delay for queuing or transmission collision). Node's initial energy is varied in $[0.3e, e]$. While a node has the max energy e, its wake-up rate is set to be 10; if its energy drop by $0.1e$, its wake-up rate decreases by 1correspondinglly.

First, we investigate the impact of network size on the end-to-end delay. The total number of nodes in the network is varied in$[60,120]$. And the network size is varied with the number of nodes correspondingly. Base stations number is fixed to be 8. As shown in Fig. 2, the performance of our algorithm is always the best because the optimization result of our algorithm is optimal or near optimal; with the increase of the network size, the time delay performances of both SPF and BPF are becoming worse, this is because, while the network size is increasing (more *k*-anycast paths are feasible to base stations), the optimization result of those algorithms are becoming more far away the optimal result.

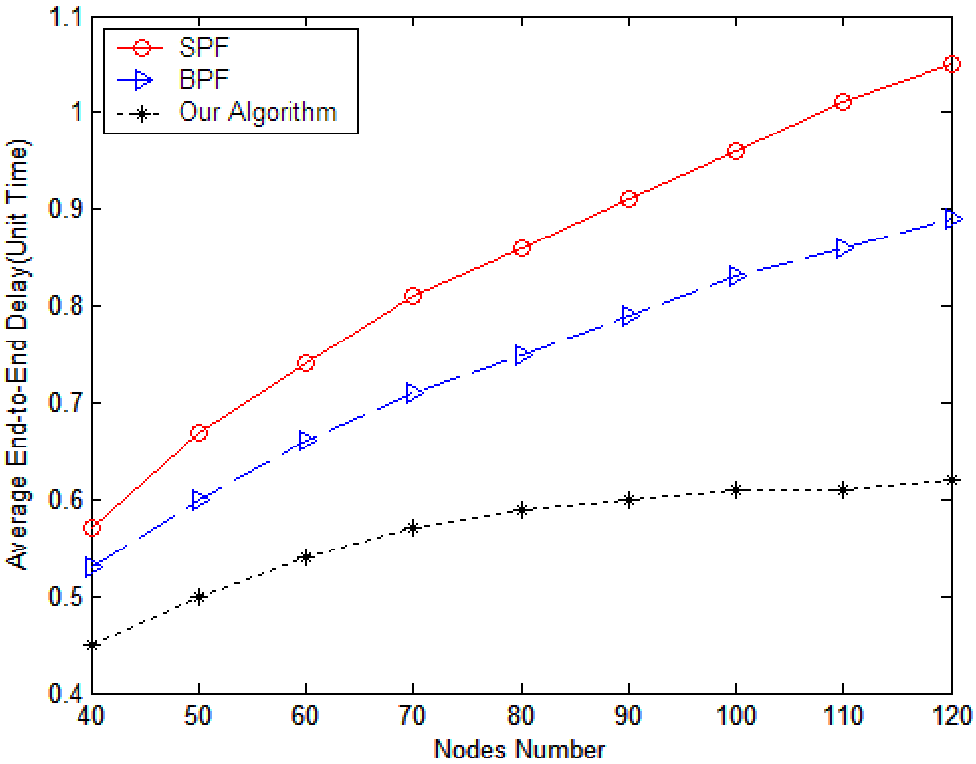

Fig 2. Nodes number and end-to-end time delay

Secondly, we investigate the impact of nodes' energy state on the end-to-end delay. For simulation the event of the energy consuming for transmission, we set 10 seconds to be a cycle time. And if a node has a transmission job in the last cycle time, the node's energy will drop by $0.1e$ in this cycle time and the node's wake-up rate drop by 1. As shown in Fig. 3, we can see that the performance of our algorithm is the best in every execution time because our optimization result is the optimal or near optimal; Since SPF select a path only care about hops(distance) not care about node's energy, the performance drop sharply while the nodes' energy are depleted which are in the shortest path.

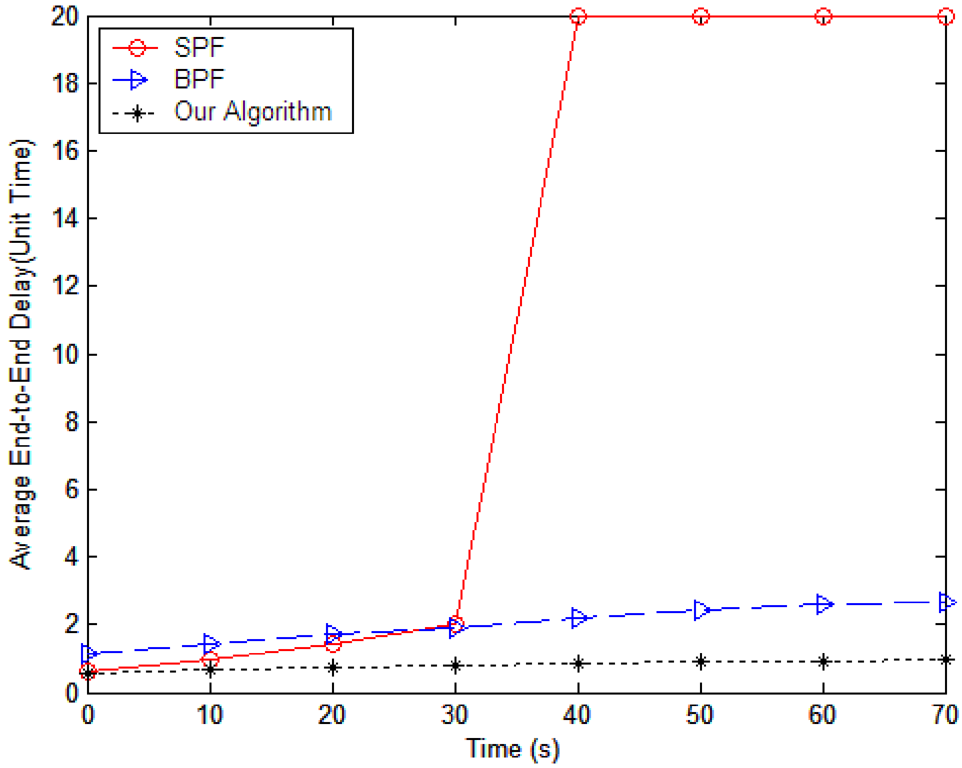

Fig 3. Execution time and end-to-end time delay

6. Conclusion

Most previous research works on finding *k*-anycast routing paths are local optimization algorithm such as SPF(Shortest Path First) and BPF((Best Path First). In this paper, we propose an optimal *k*-anycast routing algorithm for sleep-wake scheduling wireless sensor networks. In the proposed algorithm, base stations apply AODV-based multipath routing protocol to acquire *k*-anycast routing information, then genetic algorithm is applied to search the optimal *k* routing paths between nodes and bases stations. Since the proposed algorithm has the feature of global optimization ability, in contrast to previous algorithms, experiment results show that the proposed algorithm can reduce end-to-end delay more efficiently.

7. References

[1] W. Wang, X. Li, and O. Frieder, “k-Anycast Game in Selfish Networks,” Proc. 13th Int’l Conf. Computer Communication and Networks, (2004)289-294.

[2] B. Wu, and J. Wu, “K-Anycast Routing Schemes for Mobile Ad Hoc Networks,” Proc. 20th Int’l Parallel and Distributed Processing Symposium, (2006).

[3] X. Wang, *Analysis and Design of a K-Anycast Communication Model in IPv6*, Computer Communications, 31(2008)2071-2077.

[4] C. R. Dow, P. J. Lin, S. C. Chen, and et al. *An efficient anycast scheme for discovering k services in cluster-based mobile ad hoc networks*, Wireless Communications & Mobile Computing, 11(9)(2011)1287-1301.

[5] X. Wang, J. Wang, and K. Lu. *GKAR: A Novel Geographic K-Anycast Routing for Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, 24(5)(2013916-925.

[6] D. Gao, H. Lin, and X. Liu. *Routing protocol for k-anycast communication in rechargeable wireless sensor networks*, Computer Standards & Interfaces, 43(2016)12-20.

[7] J. Ren, K. Lu, F. Tang, and et al. *CAKA: a novel cache-aware K-anycast routing scheme for publish/subscribe-based information-centric network*, International Journal of Communication Systems, 28(17) (2015) 2167-2179.

[8] C.C. Hsu, M.S. Kuo, S.C. Wang, and C.F. Chou. *Joint Design of Asynchronous Sleep-Wake Scheduling and*

Opportunistic Routing in Wireless Sensor Networks, IEEE Transactions on Computers, 63(7)(2014)1840-1846.

[9] K.P. Naveen, and Anurag Kumar. *Relay Selection for Geographical Forwarding in Sleep-Wake Schedunling Wireless Sensor Networks*, IEEE Transactions on Mobile Computing, 12(3)(2013)475-488.

[10] J. Kim, X. Lin, N.B. Shroff, *Optimal Anycast Technique for Delay-Sensitive Energy-Constrained Asynchronous Sensor Networks*, IEEE/ACM Transactions on Networking, 19(2)(2011)484-497.

[11] J. Kim, X. Lin, N.B. Shroff, and et al. *Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks With Anycast*, IEEE/ACM Transactions on Networking, 18(2)(2010) 515-528.