ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 10, No. 4, 2015, *pp*. 277-284

An Anycast Routing Protocol Based on Mobile Agent for 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 01, 2015, accepted August 21, 2015)*

Abstract. The performances of most demand-driven anycast routing protocols for wireless sensor networks(WSN) are not good in terms of time delay and energy consumption. For the problem, an anycast routing protocol based on mobile agent for WSN is proposed in this paper. In this protocol, only one-hop neighbor nodes routing information is required by each sensor node, and their monitoring data needs to report to any base station by their cluster-head. Cluster heads search anycast members routing information by mobile agents and establish an anycast routing table. We then present energy consumption model of our protocol. Experiments data show that our protocol has a better performance of energy consumption in a large-scale or a high-density network..

Keywords: anycast, mobile agent, wireless sensor networks, routing protocol.

1. Introduction

Sensor nodes in wireless sensor networks (WSN) depend on limited energy capacity. So, lifetime is a key parameter for WSN. Anycast is a newly designed communication service in IPv6 which is to deliver a packet to any one member in a group of designated recipients. Applying anycast routing into a multi-hop WSN, packets generated by each sensor node may be routed to any sink nodes (i.e., base stations), which could both balance energy consumption and improve routing robustness.

For low packets rate in WSN, most proposed anycast routing protocols for WSN are based on demand-driven routing scheme[1-6], Such as Wang[1] proposed an AODV-based anycast protocol. Lenders[2] and Kserawi[3] proposed a field-based anycast routing protocol for wireless ad hoc networks, it needs periodically flooding in potential field establishment process. Jia[4] proposed a routing protocol for anycast message and they proposed a Weighted-Random Selection (WRS) approach for multiple path selection in order to balance network traffic. Demand-driven routing protocols could save much network and node resource for they needn't maintain any routing table, but they have bad performances in terms of both time delay and energy consumption.

In this paper, we propose an anycast routing protocol based on mobile agent for WSN. In this protocol, sensor nodes need only one-hop neighbor nodes routing information, monitoring data generated from each sensor node should report to their cluster-head, and the cluster- head adopts mobile agents to search or maintain routing information of anycast members. We then present energy consumption model of our protocol, experiments data show that our protocol has a better performance of energy consumption in a large-scale or a high-density networks.

2. System Model and Routing Protocol

We set WSN as a two-tier architecture[7] as shown in Fig. 1, where a cluster consists of many sensor nodes and one of them is selected as the cluster-head. While the remainder energy of a cluster-head is below the threshold, it needs a reselection according to some cluster-head selection algorithm such as LEACH[8].

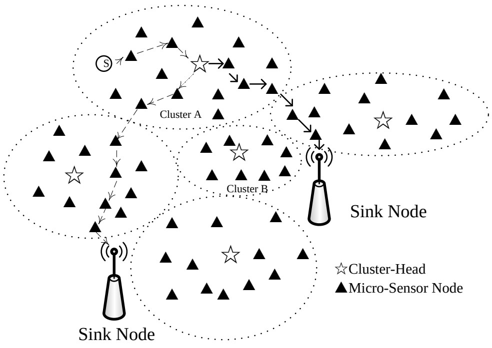

Fig. 1: WSN based on two-tier architecture.

We model WSN by a directed graph $G=(V, E)$. Where V is the set of N vertices(sink nodes and tiny sensor nodes) with a density ρ, and E is the set of m edges. All sink nodes share a same anycast address, and anycast group A (total M members and the i-th member is denoted by A_i) is denoted by $G(A)$. Assuming transmission range equal to overhearing rang, and it is denoted by d. Therefore, the number of one-hop neighbor nodes of each sensor node is n-1 (where, $n=\pi d^2\rho$).

One of responsibilities of cluster-heads is to report monitoring data generated from all sensor nodes of the cluster to any sink nodes (i.e., $G(A)$ members). For saving cluster-heads energy cost, most anycast routing protocols for WSN adopt demand-driven scheme, but they meet the problem of bad performances in term of both time delay and energy consumption. For this reason, an anycast routing protocol based on mobile agent is proposed in this paper.

2.1. Data Structure of the Protocol

Cluster-heads need to report monitoring data to any anycast member, so in our protocol they need to maintain two local information table: anycast routing table and statistics table. In our protocol, micro-sensor nodes only need local information, so they only need to maintain a local routing table only about their neighbor nodes information.

(1)Anycast routing table

Each record in anycast routing table should include unicast address and anycast address of the destination, the address of the next-hop node, routing metric, routing stack, routing weight and so on. Where, the calculation of routing metric is based on multi characters of both the anycast path and the intermediate nodes (such as time delay, distance and the remainder energy of nodes). For balancing data flow and energy consumption, we apply WRS scheme. That means, according to routing weight, the cluster-head select an anycast member randomly as its target to report the monitoring data. WRS scheme can balance both data flow and energy consumption, thus improve network lifetime.

(2) Statistics table

The structure of statistics table is as below:

$S(A_1)$	$S(A_2)$	. . .	$S(A_N)$

Where, $S(A_i)$ is a 5-tuple set $M_i\left[A, LA_i, \mu_i, \sigma_i^2, \omega\right]$. LA_i is the anycast address of A_i, u_i and σ_i^2 are the average and variance of routing metric respectively, ω is for preserving the best routing metric.

2.2. Anycast Protocol Based on Mobile Agent

In this protocol, we adopt the characters of both the vector-distance routing algorithm and the link-state routing algorithm. A sensor node needs exchange routing information only with neighbor nodes at every time interval Δt_1; a cluster-head needs to initiate M agents to search $G(A)$ members at every time interval Δt_2. A cluster-head needs 3 steps to initiate anycast routing table:

Step A: search for anycast paths between the cluster- head and $G(A)$ members.

Every defined time interval Δt_2, a cluster-head should send M mobile agents whose destination is A to search for $G(A)$ members. For shorting transversal time of agent, we give agent higher priority than monitoring data. At each node agent passes by, agent should push the node's ID into agent's stack S and record the node's ID, address, travel time and the remainder energy of the node into agent's dictionary. At each intermediate node i, agent visits local routing table, then chooses the next stop $j(j \notin S)$ according to the sequence below:

(1)Those nodes never be invited;

(2)Those nodes are $G(A)$ member, that is to say, they can be searched at node i's local routing table;

(3)Other nodes.

Which node will be selected as the next stop firstly is also determined by the remainder energy of node and the queue length of the link. We denote P_{ijA} the probability of that node j is selected as the next stop. The calculation of P_{ijA} is as follows:

$$P_{ijA} = (1-\beta)\frac{(1-q_{ij})/\sum_{j=1}^{N_i} q_{ij}}{(N_i - 1)} + \beta\frac{E_j}{E} \tag{1}$$

Where, β is a parameter for adjustment; q_{ij} is the length of the queue in node j; N_i is the number of neighbor nodes of node i; E_j is the remainder energy of node j; E is the initiate energy of node j. So E/E_j can be viewed as "transmission resistance". Selecting the path of low "transmission resistance" can avoid using the path in which the intermediate nodes are nearly exhausted energy. Thus, the value of P_{ijA} can reflect both the throughput and the rate of the remainder energy of node j.

If before arrival at the destination, agent's TTL(Time To Live) exceeds the stated boundary, then agent will be deleted. If agent arrives at A_i, it does the following process:

(a)Seek routing information left by previous agents sent by the cluster-head. If not exist, marks tag="new" in agent's dictionary, then go to step (d); otherwise, go to step (b).

(b)Compare the new path by agent and the old path by previous agents. If they are not the same, go to step (c); otherwise, marks tag="old", then go to step (d).

(c)Use routing metric to evaluate the two paths. If the old path is better, agent will be deleted, then the process is finished; otherwise marks tag="new", go to step (d).

Routing metric is calculated as below:

$$c(A_i) = \sum_{j=1}^{K} \alpha \frac{1}{t_j(A_i)} \frac{E_j}{E} \tag{2}$$

Where, α is a system parameter, K is the number of travel hops between the cluster-head and A_i, $t_j(A_i)$ is travel time during node j. So, the value of $c(A_i)$ is determined by time delay, the distance and the rate of remainder energy.

(d) In response to agent's arrival, A_i initiates a reverse mobile agent and sends it back to the cluster-head. Agent handovers its stack and dictionary to both reverse agent and A_i, then agent will be deleted. We give reverse agent the highest priority to ensure it returns to the cluster-head as soon as possible. Reverse agent should provide routing information to every clusters it passes by, and those cluster-heads consider it as their own reverse agent. This scheme is for the purpose of sharing routing information among clusters. After reverse agent arrives at the destination, it need be checked for integrality and authority. After passing the check and while tag="new", reverse agent is given authority to update anycast routing table and statistic table of the cluster-head.

Step B: update statistics table

Check the tag. If tag="new", it means a new anycast path, use Eqs.3 to update statistics table; if tag="old", it means an anycast path already recorded in anycast routing table, but the metrics $c(A_i)$ should be updated due to energy loss, time pass, etc, use Eqs.4 to update it.

$$\begin{cases} \mu_i = c(A_i) \\ \sigma_i^2 = 0 \end{cases} \tag{3}$$

$$\begin{cases} \mu_i = (1-\eta)\mu_d + \eta c(A_i) \\ \sigma_i^2 = (1-\eta)\sigma_d^2 + \eta(c(A_i) - \mu_d)^2 \end{cases} \tag{4}$$

Where, η is a parameter taken as the moving observation window. From η in Eqs.4, we know, routing information returned by the older agent, the less impact on the value of u_i and σ_i^2.

Step C: packet forwarding

According to Eqs.3 or Eqs.4, we calculate all $G(A)$ member's routing metrics u and record their routing weight in anycast routing table according to WRS[4]. When needed to report to anycast address A, the cluster-head searches the anycast routing table for the records that match A. In a series of records, we apply WRS to select one of $G(A)$ members as our routing target this time.

3. Energy Consumption Model And Analysis

In this section, we present energy consumption models of both AODV protocols (applying WRS scheme, in brief, called AODV-WRS) and our protocol, then evaluate the performances of them.

We denote $\{E_T \times k\}$ and $\{E_R \times k\}$ by the energy consumption for sending and receiving k bits packets respectively.

$$\{E_T \times k\} = E_{amp} \times d^n \times k + E_{elec} \times k \tag{5}$$

$$\{E_R \times k\} = E_{elec} \times k \tag{6}$$

Where d is the distance and n is the path loss index. In our study, we assume n=4, E_{elec}=50nJ/bit and E_{amp}=1.3pJ/bit/m^4.

Most energy consumption overhead in AODV-WRS is due to RREQ and RREP packets which mainly depend on path creation frequency r. We set r as below:

$$r = \min\{r_e, r_{max}\} \tag{7}$$

Where, r_e is monitoring event frequency and r_{max} is a given max value (system parameter).

$$\{E_T \times L_q\} r\tau(\tau-1)N^2 / C \tag{8}$$

Assuming that each cluster has average C nodes, we have total N/C cluster-heads. Every cluster-head should search base stations by flooding RREQ, and the intermediate nodes should transmit RREQ to all other neighbor nodes. Assuming node density is ρ, during node's transmit range d, each node has average $\tau=n-1$ neighbor nodes (where, $n=\pi d^2\rho$).

Each cluster-head should transmit RREQ to it's other τ neighbor nodes, and each intermediate node should transmit RREQ from one of it's neighbor nodes to the other τ-1 neighbor nodes. So, the energy consumption overhead for RREQ from every cluster-heads and thus transmission overhead in intermediate nodes is totally as below:

$$\{E_T \times L_q\} r\tau(\tau-1)N^2 / C \tag{9}$$

Where, L_q is the packet size of RREQ.

In wireless networks, while a node is sending packets, the signal is received by all nodes inside the node's broadcast radius. So, the energy consumption due to receiving RREQ in AODV-WRS is as below:

$$\{E_R \times L_q\} r\tau^2(\tau-1)N^2 / C \tag{10}$$

In AODV-WRS, all destination nodes (base station) should reply a RREP packet to cluster-heads. So, the energy consumption for sending RREP is given by

$$\{E_T \times L_p\} r \sum_{j=1}^{N/C} \sum_{i=1}^{M} h_{ij} \tag{11}$$

Where, h_{ij} is the routing hops between A_i and the cluster-head j, M is the number of anycast members, and L_p is the packet size of RREP.

In AODV-WRS, the energy consumption for receiving RREP is given by

$$\{E_R \times L_p\} r\tau \sum_{j=1}^{N/C} \sum_{i=1}^{M} h_{ij} \tag{12}$$

Each node in both AODV-WRS and MAARP should maintain it’s one-hop neighbor node’s routing information, The energy consumption for maintain local routing information is given by

$$(\{E_T \times L_s\} + \{E_R \times L_s\}\tau) r_b N \tag{13}$$

Where, L_s is the size of the exchange packet and r_b is the frequency of exchange routing information among neighbor nodes.

In conclusion, we can obtain the energy consumption for searching anycast paths in AODV-WRS as follows (omit Eqs.13)

$$E_w = \frac{(\{E_T \times L_q\} + \{E_R \times L_q\}\tau) r \tau (\tau - 1)}{C} \times O(N^2) + \frac{(\{E_T \times L_p\} + \{E_R \times L_p\}\tau) r}{C} \times \sum_{j=1}^{N/C} \sum_{i=1}^{M} h_{ij} \tag{14}$$

In the following section, we give a analysis about energy consumption of MAARP. Assuming that node density of WSN is ρ, total node number is N, and physical distance of one hop is d, we can obtain the network diameter (hop numbers) D is as below:

$$D = 2\sqrt{N / \pi\rho} \big/ d \tag{15}$$

In this paper, we consider the average hops traveled by mobile Agent for searching anycast members is approximately D. So, we can obtain the energy consumption for Agent's travelling as below.

$$\sum_{i=1}^{N/C} r_{ai} \{E_T \times L_a\} D \tag{16}$$

Where, L_a is the packet size of Agent and r_{ai} is the Agent transmit frequency of cluster head i.

In our protocol, the frequency of sending mobile agents is not the same for each cluster. In Fig.1, we can see that, cluster B has a better geography location than cluster A. That is to say, in contrast to cluster A, agents from cluster B can more easily get to sink nodes. Moreover, in comparing with cluster A, cluster B has more probability to get routing information from other agents which pass by cluster B.

We denote $(1/h_{jiA})^{\gamma}$ by the potential field value of sink node j caused by cluster-head i (Where, h_{jiA} is the hops between cluster-head i and A_j, γ is a system parameter). Thus, cluster-head i obtains the total potential field value T_i from all $G(A)$ members. T_i is given by

$$T_i = \sum_{j=1}^{M} (h_{jiA})^{\gamma} \tag{17}$$

The cluster of high field value has more advantage to get anycast path routing information. Moreover, along with the network scale grows, the advantage become more and more obvious. So, we should give a penalty to the cluster of high field value. We set r_{ai} as follows:

$$r_{ai} = \frac{r_t}{\theta T_i} \log_2 (1 + M) \tag{18}$$

Where r_t is the standard frequency, r_t and θ are system parameters. T_i *is* a penalty parameter for those clusters have a better geography location.

The energy consumption for receiving agent is as follows:

$$\sum_{i=1}^{N/C} r_{ai} \{E_R \times L_a\} (n-1)^2 R^2 M \tag{19}$$

We consider the energy consumption due to reverse agent is approximately the same with due to agent. So, in conclusion, we can obtain the energy consumption in our protocol(not include Eqs.17).

$$E_a = \left(\frac{\{E_T \times L_a\}}{n-1} + \{E_R \times L_a\}(n-1)\right) 2M \times O\left(N \sum_{i=1}^{N/C} r_{ai}\right) \tag{20}$$

Although L_a in our protocol is larger than L_q, in Eqs.18, 19, and 25, we can see that, the order of energy consumption magnitude of our protocol (O(N^2)) is less than SSP and WRS(O(N^3)). Thus, along with the network scale grows, the advantage of our protocol becomes more and more obvious. Moreover, from the analysis model we can see that, while n become larger, that is, the network density ρ grows, the performance of our protocols is better than SSP and WRS.

In this paper, the Agent transmit frequency of each cluster head is different. As figure 3.7 shown, we can see that cluster B is easier than cluster A to find a base station, and Agent from other clusters have more chance to pass by cluster B and inform it's routing information. We consider that base station j give cluster head i potential fields, and the value is $(1/h_{jiA})^{\gamma}$. Where, h_{jiA} is the hop number from cluster head i to A_j, and γ is a empirical parameter. So, cluster head i get the value of potential fields from $G(A)$ is as below:

$$T_i = \sum_{j=1}^{M} (1/h_{jiA})^{\gamma} \tag{21}$$

Clearly, the cluster-head which having a bigger fields value has more chance to acquire routing information. Moreover, with the enlargement of the network scale, the chance is still increasing. So, we can give the cluster-head which having a bigger fields a smaller Agent sending frequency (r_{ai}). We set r_{ai} as below:

$$r_{ai} = \frac{r_t}{\theta T_i} \log_2(1+M) \tag{22}$$

Where, r_t is a standard frequency; T_i *is* a penalty parameter for those clusters have a better geography location.

The energy consumption for receiving mobile Agent is calculated as below:

$$\sum_{i=1}^{N/C} r_{ai}\{E_R \times L_a\}\tau D \tag{23}$$

Reverse mobile agents will be created only mobile agents successfully arrive at destination. Assuming that, the arrival rate is α, the energy consumption for searching routing in MAARP can be calculated as below:

$$E_a = \frac{(1+\alpha)\{E_T \times L_a\}}{d\sqrt{\pi\rho}} + \frac{\{E_R \times L_a\}\tau}{d\sqrt{\pi\rho}} \times O(\sqrt{N}\sum_{i=1}^{N/C} r_{ai}) \tag{24}$$

For mobile Agent need carry data dictionary, the packet size of L_a is larger than both L_q and L_p. However, from Eqs.24, we can see that, the order of magnitude is different (MAARP is O($N^{\frac{3}{2}}$), and AODV-WRS is O(N^2)). So, MAARP will hold the advantage while the network size is increasing. Furthermore, while network density ρ is increasing, that means, the number of average neighbor nodes τ ($\tau=\pi d^2\rho$-1) is increasing, in comparison with AODV-WRS's order of magnitude(O(τ^3)), MAARP is less (the order of magnitude of the first item in Eqs.24 is O($\tau^{-1/2}$), and the second item is O($\tau^{1/2}$)). So, in comparison with AODV-WRS, MAARP hold the advantage while the network density is increasing.

4. Experiments and Analysis

The problem

In this section, we take AODV-WRS as the comparison algorithm and discuss the performance of AODV-WRS. We set WSN in a rectangle area. The parameters are shown in table.1.

Table.1:Parameter setting in WSN

ρ	0.005	L_q	192b	C	8
d	40m	L_p	160b	r_e	0.12Hz
M	4	L_a	1024B	r	3Hz

We set monitoring event rate r_e=0.12HZ, vary nodes number N(30-210) and the rectangular area correspondingly (keep ρ=0.005). The result of the energy consumption for searching route in AODV-WRS and MAARP are shown in Fig.2.

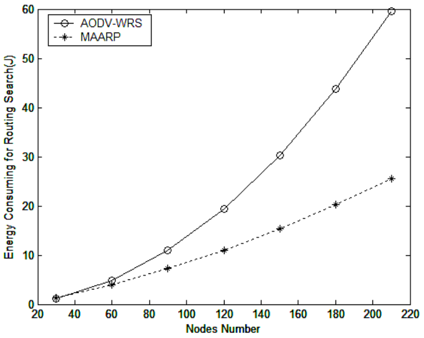

Fig. 2: Node number and energy consumption (ρ=0.005).

Fig.2 show that, while the network scale is not large, the energy consumption of MAARP is even higher than AODV-WRS, the fist reason is the size of L_a in MAARP is larger than L_q and L_p; and the second reason is that searching route rate r_t (3Hz) in MAARP exceeds AODV-WRS (0.12Hz). But, while the network scale grows, for the order of magnitude of MAARP (O($N^{\frac{3}{2}}$)) is less than AODV-WRS (O(N^2)), experiment data show that the performance of MAARP is better than AODV-WRS.

We keep nodes number N=210, and vary the network density ρ=0.001~0.005(vary network area correspondingly). The energy consumption in AODV-WRS and MAARP are shown in Fig.3.

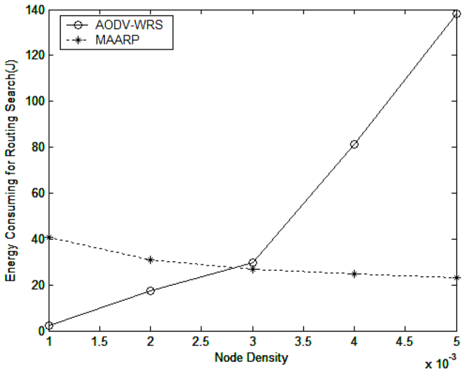

Fig. 3: Node number and energy consumption (ρ=0.005~0.035).

Fig.3 show that, while the network scale grows, node density ρ grows correspondingly, the energy consumption for receiving grows sharply. Thus, the disadvantage of demand-driven routing scheme becomes obvious. For we don’t use flooding scheme in our protocol, in comparing with demand-driven routing protocols, experiment data show that our protocol has a better performance.

In contrast to single path scheme in SSP, It needs to maintain multi anycast routing paths in WRS scheme which leads to more energy consumption. But on of advantages of WRS is that it can balance data flow and energy consumption10. We only discuss the total system energy consumption, not to discuss the character of energy balancing in the above experiments. So, our experiments data don’t reflect the advantage of energy balancing in both on-demand driven anycast routing protocols based on WRS scheme and our protocol (also use WRS scheme), this will be our later research work.

5. Conclusion

In this paper, we propose an anycast routing protocol based on mobile agent for WSN in this paper. In our protocol, cluster heads are responsible for searching anycast members routing information by mobile agents and establish an anycast routing table. Energy consumption models of demand-driven anycast routing protocol and our protocol are also presented, experiments data show that in contrast to demand-driven anycast routing protocols, our protocol has a better performance of system energy consumption in a large-scale or high-density networks.

6. References

[1] J. Wang, Y. Zheng, and W. Jia, *An Anycast Protocol Based AODV in MANET*, Proc. of the14th IEEE Conference on PIMRC, pp. 221-225, 2003.

[2] V. Lenders, M. May, and B. Plattner, *Density-Based Anycast: A Robust Routing Strategy for Wireless Ad Hoc Networks*, IEEE/ACM Transactions on Networking, 2008, 16(4): 852-863.

[3] M. Kserawi, S. Jung, D. Lee, J Sung, and J.-K.K. Rhee, *Multipath Video Real-Time Streaming by Field-Based Anycast Routing*, IEEE Transactions on Multimedia, 2014, 16(2): 533-540.

[4] W. Jia, D. Xuan, W. Tu, L. Lin, and W. Zhao, *Distributed admission control for anycast flow*, IEEE Transactions on Parallel and Distributed Systems, 2004,15(8):673-686.

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

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

[7] Y. T. Hou, Y. Shi, and H. D. Sherali, *Optimal Base Station Selection for Anycast Routing in Wireless Sensor Networks*, IEEE TRANSACTIONS ON Vehicular Technology, 2006, 55(3):813-821.

[8] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, *Energy-Efficient Communication Protocols for Wireless Microsensor Networks*, Proc. of the 33th IEEE Conference on HICSS, pp. 223-233, 2000.

[9] L. Xiao, S. Boyd, and S. J. Kim, *Distributed Average Consensus with Least-Mean-Square Deviation*, Journal of Parallel and Distributed Computing, 2007, 67(1):33-46.