Energy Efficient Switch-Based Packet Forwarding for Low Duty-Cycle Wireless Sensor Networks

Xinming Zhang, *Member, IEEE*, Lei Tao, and Fang Zhou

***Abstract*—This letter studies the problem of energy efficient routing in multihop wireless sensor networks (WSNs) with low duty cycle, where a switch-based forwarding scheme is employed. We consider the unreliable end-to-end packet transmissions and propose a routing metric, expected efficient energy consumption (EEEC), to reflect the energy consumption incurred in the routing process. We give an analytic model of the proposed metric and devise distributed algorithms which can find an EEEC-optimal relay sequence for the forwarders under a delivery ratio constraint. Simulation results illustrate the benefit of our proposed metric and algorithms over the state-of-the art.**

***Index Terms*—Energy efficient routing, relay selection, sleep-wake scheduling, wireless sensor networks.**

I. INTRODUCTION

ENERGY efficiency is crucial for designing routing protocols in wireless sensor networks (WSNs) with limited battery supplies. Duty-cycled wireless sensor networks (DC-WSNs) permit nodes to go to dormant state with no packets to transmit to save energy. And *switch-based packet-forwarding* schemes have been proposed for DC-WSNs, where a forwarder selects a set of neighboring nodes as its potential relays instead of a deterministic next hop. Delay and energy can be significantly reduced with multiple relays providing more chances for the forwarder to forward packets.

There are two critical issues for energy efficient routing under switch-based forwarding framework: metric design and optimal relays selection. Yu and He [1] defined energy efficiency as the number of transmissions and proposed Expected Energy Consumption (EEC) to measure the energy consumption to deliver a packet to the sink. Ghadimi *et al.* [2] proposed Expected Duty Cycled wake-ups (EDC) to indicate the radio-on time to transmit a packet to the sink, which is regarded as a metric of delay as well as energy. However, on the one hand, as pointed out in [3], we find current studies only considered the energy of successful transmissions to the sink while overlooking the possibility of packets dropped on the way to the sink in a network with unreliable links and nodes. On the other hand, the relay selection algorithm of [1] is a greedy one and that of [2] is proposed under a simplified model, which may make their selected relays less than optimal. In this letter, we propose Expected Efficient Energy Consumption (EEEC) to build a more accurate analytic model to estimate the energy consumption incurred in the routing process, and formulate the relay selection problem under end-to-end delivery ratio constraint as a constrained dynamic programming problem. Then we use a subgradient method under lagrangian relaxation to solve the problem.

II. METRIC DESIGN

A. System Model

We first introduce the underlying protocols adopted: receiver-initiated mechanism and switch-based forwarding scheme. We use receiver-initiated mechanism [4] to exchange messages between nodes. We adopt it under sleep-wake schedule aware case, where nodes can derive when a neighbor wakes up (namely neighbor discovery latency) according to the local sleep-wake schedule information of neighbors. Each receiver broadcasts a beacon frame every time it wakes up to declare that it is ready to receive a data frame from potential senders. It stays awake for t^{on}, and goes back to sleep if no response is detected, otherwise it stays awake to receive the data and send back another beacon as an ACK. On the other hand, a sender waits for the beacon from the receivers. If no receiver is awake, it goes back to sleep and wakes up slightly before the next time the receiver wakes up. Upon receiving the beacon, the sender starts its data transmission immediately, which will be acknowledged by the receiver.

Fig. 1 depicts the basic principles of receiver-initiated mechanism. T is the sleep-wake period of R. The sender S actually wakes up t_A earlier than the receiver R in case of clock drift between them in real-world environment. t_D and t_B are the transmission latency of a data packet and a beacon packet respectively. S fails to transmit the data packet to R at the first try, it does not get an ACK in time and then goes back to sleep early, until R wakes up again for S to successfully transmit the packet. In terms of deducing the neighbor discovery latency, reference [5] gives a solution: assume the periodic sleep-wake schedule of node i is $W_i = (t_i^0, T_i)$, where t_i^0 is the first time that i wakes up and T_i represents its sleep-wake period, we can express the discovery latency of i to j at time $t_i^k = t_i^0 + kT_i$ as:

$$\delta_{i,j}^{t_i^k} = T_j - (kT_i - \delta_{i,j}^{t_i^0}) \; mod \; T_j, \; k > 0. \tag{1}$$

And an additional package exchange process is used to measure $\delta_{i,j}^{t_i^0}$.

Switch-based forwarding is a kind of opportunistic routing protocols designed for WSNs with unreliable links. A series of

---

Manuscript received December 15, 2015; revised January 24, 2016 and February 16, 2016; accepted March 6, 2016. Date of publication March 9, 2016; date of current version May 6, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61379130 and Grant 61073185, and in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University under Grant 2014D06 and Grant 2010D15. The associate editor coordinating the review of this paper and approving it for publication was Y. Chen.

The authors are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: xinming@ustc.edu.cn; taolei@mail.ustc.edu.cn; zffang@mail.ustc.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2016.2539939

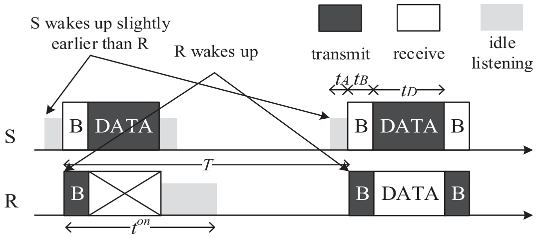

Fig. 1. RI-MAC with sleep-wake schedule awareness.

neighboring nodes called relay set are carefully chosen by the forwarder as its next hops to relay packets. Switch-based forwarding allows a dynamic, any-path routing via opportunistic relay selection, where all the nodes in relay set have the opportunity to relay packets from the forwarder. The forwarder takes turn to (re-)transmit the data packets to each node in its relay set, until one successfully receives the packet to become a new forwarder. Due to the switch-and-transmit property, it is called switch-based packet-forwarding.

B. EEEC and Its Analytic Model

In this subsection, we introduce the energy efficient routing metric EEEC and its analytic model. The energy cost to deliver a packet for the transceiver is made up of three components: the RX/TX, idle listening and sleep energy. Due to the extremely low sleep power, we neglect the energy of sleeping and only consider the energy of idle listening and RX/TX in the analysis. We propose EEEC to describe the energy cost to achieve energy efficient routing. The EEEC routing metric is an indicator of energy consumption in the packet transmission process along the routing path. Thus, choosing a route with low EEEC leads to reduced energy consumption.

We consider a DC-WSN as an undirected graph $G = (V, E)$, where V is the set of all sensor nodes in the network, and $E = \{e_{ij} \mid i, j \in V\}$ is the set of links between vertices. We say node u and node v are neighbors of each other if $e_{uv} \in L$. We use p_{uv} to denote the probability of a packet to successfully transmit through the link e_{uv}, which can be estimated by Collection Tree Protocol (CTP) [6]. We represent the relay set of node i as R_i. We assume R_i is fixed for the moment and will discuss in section III how to perform relay selection. Once R_i is determined, i orderly transmits data packet to nodes in R_i based on their wake-up time, until one of them successfully receives the packet. However, in real-world network settings, a forwarder can not *immoderately retransmit a packet* because that causes increased one-hop delay and energy consumption at sender side [1]. A predefined retransmission count limit also prevents a sender from retransmitting one packet too many times. Thus, only a finite number of nodes in the relay set have the opportunity to become the forwarder. We call the finite node sequence a relay sequence and denote it as RS_i.

Let EEC_i be the energy required for a packet transmission from i towards the sink along the routing path with its relay sequence RS_i. We consider not only the energy consumption when the current sender successfully transmits the packet to one of its forwarders in RS_i, which is expressed as EEC_i^s, but also the one-hop energy consumption for a failed forwarding process at the current sender, which is EEC_i^f. We give its expectation as

$$E[EEC_i] = E[EEC_i^s] + E[EEC_i^f]. \tag{2}$$

To compute $E[EEC_i^s]$, we use N_i to denote that the transmission successes at the N_ith relay node and define $p(l)$ as the probability of $N_i = l$. We have

$$p(l) = \prod_{k=1}^{l-1}(1 - p_{i,k})p_{i,l},\ 1 \le l \le K. \tag{3}$$

Note that we use the index of the node in RS_i to represent the node for simplicity here and hereinafter. K represents the size of relay sequence RS_i.

The single-hop energy consumption with $N_i = l$ is the sum of the energy cost at the sender plus the energy cost at the receiver, which is represented as $ECS_{i,l}$ and $ECR_{i,l}$ respectively. We let P_{il}, P_{tx} and P_{rx} to represent the idle listening, transmitting and receiving power respectively, then $ECS_{i,l}$ can be expressed as

$$ECS_{i,l} = l(P_{il}t_A + P_{tx}t_D). \tag{4}$$

The data packet size is assumed to be fixed here and the overhead of control packets is not considered in the analysis.

Before computing $ECR_{i,l}$, we first consider a fact that in DC-WSNs with receiver-initiated mechanism, when no packets are transmitting, nodes periodically wake up and idlely listen to the channel for a period of time. The energy consumption of this wake-up process is predetermined and should be excluded from $ECR_{i,l}$ in designing EEEC. Back to our analysis, the receiver receives the data packet for t_D to find a broken packet due to path losses in turns $1, 2, \ldots, l-1$ and finally decode the packet successfully at turn l. We use $ECR_{i,l}^*$ to denote the extra energy consumption incurred in routing for the receivers instead of $ECR_{i,l}$, which can be approximated as

$$\begin{aligned} ECR_{i,l}^* &= \sum_{k=0}^{l}\left(P_{il}(t_k^{on} - t_D - t_B) + P_{rx}t_D - P_{il}(t_k^{on} - t_B)\right) \\ &= l(P_{rx} - P_{il})t_D. \end{aligned} \tag{5}$$

The multi-hop energy consumption with $N_i = l$ is EEC_l, indicating the energy cost-to-go to forward this packet when node l becomes the forwarder. Now we can write $E[EEC_i^s]$ as

$$E[EEC_i^s] = \sum_{l=1}^{K} p(l)(ECS_{i,l} + ECR_{i,l}^* + E[EEC_l]). \tag{6}$$

When all the attempts to transmit the packet to the relay sequence fail, we call it a failed forwarding process and use EEC_i^f to denote the energy consumption in the process. Clearly, the energy consumption at the sender equals to $ECS_{i,K}$ and, the extra energy consumption at the receivers equals to $ECR_{i,K}^*$. $E[EEC_i^f]$ can be expressed by

$$E[EEC_i^f] = \prod_{l=1}^{K}(1 - p_{i,l})(ECS_{i,K} + ECR_{i,K}^*) \tag{7}$$

Finally, we can use equation (2) to calculate $E[EEC_i]$:

$$E[EEC_i] = \sum_{l=1}^{K} p(l)(l\Omega + E[EEC_l]) + \prod_{l=1}^{K}(1 - p_{i,l})K\Omega, \tag{8}$$

where $\Omega = P_{il}t_A + (P_{tx} + P_{rx} - P_{il})t_D$.

To this end, we derive the formal expression of EEEC, which actually depicts all the possible routes in anycast routing towards sink, and use a weighted average under switch-based forwarding model to estimate the overall energy consumption in the packet transmission process. In the proposed analytic model, we consider the energy of idle listening and RX/TX, as well as the energy consumption when multi-hop packet transmission fails on the way to the sink. In comparison, EEC [1] simply proposes the number of transmissions as the energy metric and does not consider the energy consumption of failure transmission. Thus, EEEC provides more accurate estimation on energy consumption.

III. EEEC-OPT RELAY SEQUENCE SELECTION

The relay sequence RS_i is assumed fixed in the analytic model. In this section, we propose an algorithm for each node to select relays that minimizes EEEC. We only consider the delay-constrained situation in this letter, where a time limit T_{th} is preset as the longest time for a forwarder to hold a packet in one-hop transmission. The relay sequence candidates for i consists of the sequence of all its neighbors waking up before T_{th}, ordered by their wake-up time. We write it as $C_i^{can} = (c_1, c_2, \ldots, c_M)$ and the problem turns into selecting a subset RS_i from C_i^{can} to minimize the EEEC value.

One naive method is to exhaust all the subsets of C_i^{can} to find one that has the minimum EEEC value, whose complexity is $O(2^D)$ (D indicates the number of neighbors), which makes it not a feasible solution. We adopt a dynamic programming method [7] to solve the problem. We first prove that the relay sequence selection problem exhibits the *Optimal Substructure* property. We make assumptions that the optimal relay sequence of a node sequence $X = (c_{s_1}, c_{s_2}, \ldots, c_{s_x})$ is $A = (c_{u_1}, c_{u_2}, \ldots)$. If there exists a node sequence $Y = (c_{t_1}, \ldots, c_{t_y}, c_{s_1}, c_{s_2}, \ldots, c_{s_x})$, whose optimal relay sequence is $B = (c_{v_1}, c_{v_2}, \ldots)$, where $X \subseteq Y$ but $A \nsubseteq B$. We define $c_{v_m} \in B$ as the last node ahead of $c_{s_1} \in Y$, thus B can be divided into $B_0 = (c_{v_1}, \ldots, c_{v_m})$ and $B_1 = (c_{v_{m+1},\ldots})$. C_B, C_{B_1} and C_A are used to denote the EEC value with relay sequence B, B_1 and A respectively. From equation (8) we can express C as

$$C_B = \prod_{l=1}^{m}\big((1 - p_{v_l})C_{B_1} + p(l)E[EC_{v_l}]\big) + \sum_{i=1}^{m}\prod_{j=1}^{i-1}(1 - p_{v_j})\Omega. \tag{9}$$

We can replace B_1 with A in B, which equals to replacing C_1 with C_A in cost equation (9), thus decreases C_B because A is superior to B_1. It is conflict with the assumption that B is an optimal relay sequence. Thus, there must be $A \subseteq B$. The Optimal Substructure property is then satisfied and the optimality of the dynamic programming approach is proved. Then we adopt a backward approach to try to add nodes one by one from C_i^{can} to RS_i. If adding a node leads to a smaller EEEC value, then including it into RS_i, else we skip to the next one. Since the last node in the optimal substructure is not determined, we need to try every node in C_i^{can} as the initial last node. We call the above-mentioned algorithm EEEC-Minimum Relay Selection.

Algorithm 1. EEEC-OPT Relay Selection With EDR Bound

```
1: $n = 0,\ \lambda^0 = 0,\ edr^0 = 0$
2: while $edr^n < EDR_b$ do
3:     for $l$ from 1 to $M$ do
4:         $\overline{E_l} = E[EEC_l] - \lambda^n E[DR_l]$
5:     end for
6:     Perform $\overline{\text{E}}$-Minimum Relay Selection to get $RS_i$
7:     $n = n + 1$
8:     $\lambda^n = \lambda^{n-1} + \gamma^{n-1}(EDR_b - edr^{n-1})$
9:     Compute $edr^n$ using (10)
10: end while
11: Compute $E[EEC_i]$ using (8)
12: return $E[EEC_i]$,$RS_i$
```

However, if there is no constraint on the selection, the algorithm in terms of minimizing energy prefers to include only few nodes into the relay sequence, leading to extremely low EDR (end-to-end delivery ratio). Therefore an EDR bound EDR_b is necessary to guarantee the reliability of routing. We formulate the problem of minimizing EEEC with EDR bound as

$$E^* := \min_{RS_i} E[EEC_i], \quad s.t.\ E[DR_i] \geq EDR_b. \tag{O}$$

$E[DR_i]$ indicates the EDR value of node i, which can be calculated by

$$E[DR_i] = \sum_{l=1}^{K} p(l)E[DR_l], \tag{10}$$

according to [1]. We adopt a lagrangian relaxation to solve this constrained dynamic program. The relaxed problem is considered as:

$$E(\lambda) := \min_{RS_i} E[EEC_i] + \lambda(EDR_b - E[DR_i]). \tag{R}$$

Since for the solutions that satisfy problem (O) and all $\lambda \geq 0$, $E(\lambda)$ is a lower bound of E^*, we would like to find the tightest of all bounds to narrow the gap between $E(\lambda)$ and E^*, namely we search for:

$$E_D := \max_{\lambda} E(\lambda). \tag{D}$$

A subgradient optimization method [8] is adopted here to solve the dual problem (D). The formal description of the method is presented in Algorithm 1.

λ^n and edr^n denote the lagrangian multiplier and the EDR at iteration n respectively. $\overline{\text{E}}$ indicates the new EEEC value after relaxation. At each iteration, edr^n is checked: if it does not meet the requirement of the EDR constraint, we update the value of $\overline{\text{E}}$ with relaxed functions (line (3)-(5)) and perform $\overline{\text{E}}$-Minimum Relay Selection (line (6)). The new EDR is computed and the multiplier is updated after the iteration n increases.

Finally a relay sequence that satisfies EDR_b is obtained, we calculate the EEEC value (line 11). The complexity of the proposed algorithm is $O(DQ)$, where Q indicates the number of iterations. According to [8], the stepsize γ^n should satisfy $\lim_{n\to\infty}\gamma^n = 0$ and $\sum_{n=0}^{\infty}\gamma^n = \infty$ to guarantee the convergence of process E_D, and we choose γ^n as $1\big/lg(10+n)$ in this letter.

IV. PERFORMANCE EVALUATION

We evaluate the performance of the proposed metric and routing policy in this section. A monitoring WSN is simulated where 500 sensor nodes are randomly deployed over a $300 \times 300m^2\}$ area. The sink node is in the middle of the area and each node regularly generates a report to the sink. The size of a data frame and a beacon frame is set to 100 and 30 Bytes respectively. The duration t_A is set to 20ms. We refer to the CC2420 chipset and set the power of transmitting, receiving and idle listening to 52.2mW, 59.1mW and 59.1mW respectively at 0dbm with voltage of 3V. Other parameters also follow CC2420 if not special specified. We set the reference duty cycle length T to 1s, which is also the value of the one-hop retransmission time limit T_{th}. Each node decides its duty cycle length randomly from $[0.5T, 1.5T]$. The maximum iteration number is empirically set to 60. If the iteration number exceeds 60 the algorithm 1 will stop and return the best result by far. All the results are obtained from the average of 20 different network topologies. We compare the performance of the following algorithms: **EEC-greedy**: energy efficient routing scheme proposed in [1], which adopts a greedy method to add a node that yields a minimal EEC value into the relay sequence at each stage until the EDR bound is satisfied.

EEEC-greedy: the proposed EEEC metric with a similar greedy relay selection policy as EEC-greedy.

EEEC-opt: the proposed EEEC metric with the algorithm "EEEC-OPT relay selection with EDR bound".

In Fig. 2 and Fig. 3, we compare the end-to-end delivery ratio and the normalized energy consumption under different EDR thresholds. We observe that in terms of end-to-end delivery ratio, the EDR threshold is basically achieved by all 3 algorithms where EEEC-opt is always above the threshold line while the other two algorithms are sometimes slightly below the threshold line. This is because in EEEC-opt relay selection, the forwarder only includes a front node into the relay sequence candidate whose EEEC value strictly lower than its current EEEC value at each stage (this can be derived by Eq. (8)), which in fact prevents routing loops. While for the greedy algorithms, nodes are added with a random order and such property does not hold any longer thus potential routing loops reduce their real EDRs. We also observe that both EEEC-greedy and EEEC-opt outperform EEC-greedy in terms of energy efficiency. EEEC-greedy outperforms EEC-greedy because EEEC-greedy estimate the energy more accurate instead of modeling the energy simply with transmission numbers. And the performance gap between EEEC-opt and EEEC-greedy is that the former adopts a more-superior relay selection scheme. We also observe that as EDR threshold increases, the normalized energy consumption with EEEC-based algorithms also increases. In order to satisfy the EDR constraint, neighbors which are not selected as relays before for their high EEEC values will be included into the relay sequence. As a result, the EEEC value at each forwarder increases and the energy consumption of the network increases. EEEC-greedy is more complexity efficient while EEEC-opt is more energy efficient requiring a bit more computational resources. According to our simulation results, the average iteration number of EEEC-opt is 30.

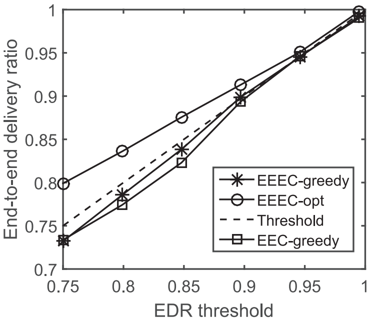

Fig. 2. End-to-end delivery ratio vs. EDR thresholds.

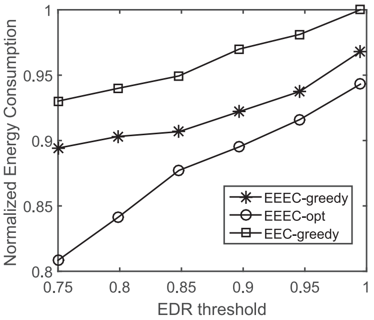

Fig. 3. Normalized energy consumption vs. EDR thresholds.

V. CONCLUSION

In this letter, we focused on the energy efficiency of routing in low duty-cycle WSNs. Our proposed metric EEEC reflects the energy in the routing process with consideration of the unreliability of links. An optimal problem has been suggested to determine the relays for the forwarder nodes while satisfying the requirements for end-to-end delivery ratio, and an algorithm to find the optimal solution has been proposed. The simulation results shows that the proposed metric and algorithm can improve the energy efficiency of routing.

REFERENCES

[1] G. Yu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 12, pp. 1741–1754, Dec. 2011.

[2] E. Ghadimi *et al.*, "Opportunistic routing in low duty-cycle wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 10, no. 4, p. 67, 2014.

[3] J. Wang *et al.*, "QoF: Towards comprehensive path quality measurement in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 1003–1013, Apr. 2014.

[4] X. Fafoutis *et al.*, "Receiver-initiated medium access control protocols for wireless sensor networks," *Comput. Netw.*, vol. 76, pp. 55–74, 2015.

[5] S. Lai and B. Ravindran, "Least-latency routing over time-dependent wireless sensor networks," *IEEE Trans. Comput.*, vol. 62, no. 5, pp. 969–983, May 2013.

[6] O. Gnawali *et al.*, "Collection tree protocol," in *Proc. ACM Sensys*, 2009, pp. 1–14.

[7] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA, USA: Athena Scientific, 2000.

[8] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.