Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 460506, 9 pages
http://dx.doi.org/10.1155/2015/460506

Research Article

A Gradient-Assisted Energy-Efficient Backpressure Scheduling Algorithm for Wireless Sensor Networks

Zhenzhen Jiao,[1,2] Rui Tian,[3] Baoxian Zhang,[1] and Cheng Li[4]

[1] *Research Center of Ubiquitous Sensor Networks, University of Chinese Academy of Sciences, Beijing 100049, China*
[2] *China Academy of Information and Communications Technology, Beijing 100033, China*
[3] *Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, Beijing 100024, China*
[4] *Faculty of Engineering and Applied Science, Memorial University, St. John's, NL, Canada A1B 3X5*

Correspondence should be addressed to Cheng Li; licheng@mun.ca

Received 1 July 2015; Revised 8 October 2015; Accepted 20 October 2015

Academic Editor: Lijun Qian

Backpressure based scheduling has revealed remarkable performance in wireless multihop networks as reported in a lot of previous work. However, its lack of consideration on energy use efficiency is still an obstacle for backpressure based algorithms to be deployed in resource-constrained wireless sensor networks (WSNs). In this paper, we focus on studying the design of energy efficient backpressure based algorithm. For this purpose, we propose a gradient-assisted energy-efficient backpressure scheduling algorithm (GRAPE) for WSNs. GRAPE introduces a new link-weight calculation method, based on which gradient information and nodal residual energy are taken into account when making decisions on backpressure based transmission scheduling. According to the decisions made by this new method, packets are encouraged to be forwarded to nodes with more residual energy. We theoretically prove the throughput-optimality of GRAPE. Simulation results demonstrate that GRAPE can achieve significant performance improvements in terms of energy use efficiency, network throughput, and packet delivery ratio as compared with existing work.

1. Introduction

Backpressure algorithm was proposed by Tassiulas and Ephremides in [1] and it has been proven to be throughput optimal. Backpressure algorithm is purely queue length based and it works in a way such that packet transmission scheduling decisions are made based on queue backlog differentials between neighboring nodes. Recently, design of efficient backpressure algorithms has attracted a lot of attention and much work has been done in this area. On one hand, backpressure based algorithms have many remarkable advantages; for example, they can achieve adaptive resource allocation and support to dynamic stateless load-aware routing and scheduling and simplicity. On the other hand, they also have some deficiencies such as large end-to-end (E2E) delivery delay, high queueing complexity, and centralized computation mode, which largely affect their usage in practice. Recently, much progress (e.g., [2–20]) has been made for supporting efficient and practical backpressure based scheduling and routing in various networks and application scenarios. However, how to enable practical backpressure based scheduling in a wireless sensor network (WSN) is still far from being well studied.

WSNs are often considered to be resource constrained where energy use efficiency is in general a great design concern for network protocols to be useful in such networks. Although existing work (e.g., [2, 4, 10]) has made certain progresses in enabling efficient backpressure based scheduling in a WSN, lack of consideration on energy use efficiency is still a big issue in their usage. To ease the understanding of the issue that backpressure based scheduling faces in this aspect, here, let us take a look at the operational process taken by classical backpressure scheduling algorithm. In the classical backpressure algorithm, per-flow queues (or per-destination queues as used in some work) are required to be maintained for each node in the network. At each time slot, the algorithm works to activate a set of noninterference links whose link-weights yield a global maximal sum to transmit packets. The link-weight is assigned to be the maximal flow-weight and the flow-weight is equal to the differential of corresponding flow's queue backlogs between the link's two endpoints. In such a way of transmission scheduling, packets are always pushed away from network hot-spots (by the so-called back pressure), no matter whether such transmissions lead to routing detours or even loops. One advantage of such backpressure based scheduling is that the capacity of the whole network can be fully utilized. However, long E2E packet latency is often observed. Furthermore, lack of consideration on energy use efficiency when making decisions on next hop selections in such algorithms results in poor network lifetime performance.

In this paper, we focus on studying the design of energy-efficient backpressure based scheduling algorithm for WSNs. For this purpose, we propose a gradient-assisted energy-efficient backpressure scheduling algorithm (GRAPE). GRAPE introduces a new link-weight assignment method, according to which a link's weight is determined by not only the differential between its two endpoint nodes' queue backlogs but also the recipient's residual energy status as well as their gradient difference. In GRAPE, packets are encouraged to be forwarded to neighbor nodes with more residual energy and lower gradients. We present the design details of GRAPE and then theoretically prove its throughput-optimality. Extensive simulation results demonstrate that GRAPE can yield significant performance improvements in terms of energy use efficiency, network throughput, and packet delivery ratio performance as compared with existing work such as the classical backpressure algorithm [1], enhanced dynamic back-pressure routing algorithm (EDR) [3], and min-hop routing.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 presents our system model. In Section 4, we first introduce how the classical backpressure algorithm works and then introduce the motivation behind our work in this paper via some simulations. Finally, we present the design details of GRAPE and further prove its throughput optimality. Extensive simulation results are presented in Section 5. In Section 6, we conclude this paper.

2. Related Work

Recently, much progress has been made in the design of efficient backpressure based scheduling algorithms for wireless multihop networks. Existing work in this field can be divided into two types: one is aimed at reducing the path lengths and thus reducing the E2E packet delay another is aimed at improving the queueing structure kept at nodes and thus improving the scheduling performance. Next, we will briefly review typical work belonging to either type.

Some existing backpressure based algorithms/protocols (e.g., [2–6, 10]) work to reduce the chance of using long or detour routes. BCP [2] is a backpressure based data collection protocol for WSNs. In BCP, backpressure based routing decisions are made based on queue backlog differential and also estimated link rates. Furthermore, BCP uses a routing-loop-punishment factor for avoiding routing loops. Furthermore, a LIFO (Last-In-First-Out) queue structure is adopted, which can help reduce the average E2E delay. BCP demonstrates good E2E performance comparable to the well-known collection tree protocol (CTP) [21], especially for networks with mobile elements. In [3], Georgiadis et al. proposed an enhanced dynamic backpressure routing algorithm (EDR). In EDR, decisions on routing and scheduling are made by taking the hop-distance to destination into account. In EDR, for instance, a neighbor node closer to the destination node may have higher probability to be chosen as the next hop forwarder than a remote node when the former has equal or even higher queue backlog than the latter. The flow weight assignment in EDR can help reduce certain routing detours and is also helpful in reducing energy consumption to certain extent due to the preference to shorter paths. Similar strategies can also be found in [4], where several factors including link capacity and network external arrival rates are considered into the routing decision making process. In [5], Ying et al. proposed a protocol that combines backpressure algorithm and shortest-path routing, which minimizes the average path length determined by backpressure based routing and thus reduces the average E2E delay. In [6], Maglaras and Katsaros proposed a layered backpressure routing algorithm. The main idea is similar to the gradient based routing in WSNs; that is, nodes are divided into layers based on their hop distances to the sink node and data packets are encouraged to be sent from nodes at higher layers to nodes at lower layers. In [10], Jiao et al. proposed an anycast based backpressure scheduling algorithm for WSNs, in which anycast based backpressure scheduling is realized in the RTS-CTS handshaking process among neighbor nodes in a localized manner. In this algorithm, packets are restricted to be forwarded to neighbor nodes with lower gradients.

Some existing backpressure based algorithms/protocols (e.g., [7–9]) choose to use new queue structures to replace the commonly used per-flow or per-destination queues for reducing the queueing complexity as well as the average E2E delivery latency. In [7, 8], a novel per-neighbor queue structure was proposed. This new queue structure enables nodes to only maintain one forwarding queue for each neighbor, which exhibits low average-case E2E delay and also low queueing complexity. In [9], Ying et al. proposed a cluster based backpressure algorithm, according to which each node keeps two types of queues, that is, one for the gateway node for each destined cluster and another for nodes in the same cluster. In this way, the cluster based backpressure routing largely reduces the number of queues required to be kept at each node.

There also exist some other algorithms (e.g., [12–18]) that attempt to improve the practicality of backpressure algorithms. For example, in [12], an adaptive redundancy technique for backpressure routing was introduced, in which replicas are generated and sent as regular packets for reducing the E2E delay under low load conditions, while traditional backpressure routing is still used under high traffic load conditions. In [18], interflow network coding was introduced and integrated with backpressure scheduling, in which network coding gain is utilized for assisting backpressure based transmission scheduling and thus reducing the E2E delay. However, these algorithms often cause some overhead during their operational phase, which are not desirable for a resource constrained WSN. In this paper, we aimed at designing energy-efficient backpressure based algorithm for a WSN. To the best of our knowledge, this is the first attempt in this aspect.

3. System Model

In this paper, the WSN under study can be modeled by graph $G = (V, E)$, where V and E represent the sets of nodes and the set of links in the network, respectively. $V(G)$ consists of many sensor nodes and one sink node. Sensor nodes generate sensing data packets when they sample new data and then inject the data into the network. The sink node is the only destination of all the data packets generated by sensor nodes. We assume time is slotted, which is denoted by t.

3.1. Queue Dynamics and Stability under Classical Backpressure Algorithm. Before introducing how our algorithm works, let us first introduce how the queues in the classical backpressure algorithm in [1] evolve. The classical backpressure algorithm requires each node $a \in V(G)$ to maintain a forwarding queue for each flow traversing it. We denote the per-flow queue backlog of flow f on node a at time t by $U_a^f(t)$.

At the beginning of each time slot, external data traffic of each flow is injected into the network via the source node of the flow. For example, the dynamics of queue backlog of flow f, where $A_f(t)$ denotes the number of packets of flow f that actually arrive at the queue at flow f's source node (denoted by $b(f)$), are as follows:

$$U_{b(f)}^f(t+1) = U_{b(f)}^f(t) + A_f(t). \qquad (1)$$

Furthermore, as traffic always leaves the network layer when they reach their destination(s), for the destination of a flow f (denoted by $e(f)$), its queue backlog will always equal to zero; that is,

$$U_{e(f)}^f(t) = 0, \quad \text{for } \forall t \geq 0. \qquad (2)$$

A network's stability is defined via the dynamics of queues in the network; that is, we can call that a network is strongly stable when for all $a \in V(G)$ and $f \in F$

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[U_a^f(\tau)\right] < \infty. \qquad (3)$$

3.2. Flows and Queue Dynamics in a WSN. The status of flows in a WSN is quite different from those considered in most previous backpressure based algorithms. For example, there exists a common assumption in the study of backpressure based algorithms; that is, flows are long-lived and data sources are fixed. However, this assumption does not hold in a WSN, where each sensor node may start to generate packets or stop at any time, especially in some environment monitoring applications wherein sensors sample the environment and send collected data to the sink upon the occurrence of particular events. Furthermore, as in many cases, these sensing packets should be served equally and they all have a common destination, that is, the sink node in the network. Thus, all data packets in a WSN can be considered to belong to the same flow and managed by only using one flow-specific queue at each node (this is also straightforward from the perspective of per-destination queue structure). The source of such a flow is a node set which includes all sensor nodes in the network. As a result, we rewrite the queueing dynamics equations for a WSN as follows.

At the beginning of each time slot, external data traffic may enter the network via any sensor node. For a sensor node a, the dynamics of its queue backlog are as follows, where $A_a(t)$ denotes the number of packets that actually arrive at node a:

$$U_a(t+1) = U_a(t) + A_a(t). \qquad (4)$$

Furthermore, the sink's queue backlog will always equal to zero; that is,

$$U_{\text{sink}}(t) = 0, \quad \text{for } \forall t \geq 0. \qquad (5)$$

We call that a WSN is strongly stable when for all $a \in V(G)$

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[U_a(\tau)\right] < \infty. \qquad (6)$$

4. GRAPE: Motivation, Design, and Analysis

In this section, we first introduce how the classical backpressure scheduling algorithm works. We then bring some experimental results that motivate our work in this paper. Finally, we propose the design details of GRAPE and further prove its throughput-optimality.

4.1. Classical Backpressure Scheduling. The classical backpressure algorithm in [1] works as follows. First, it assumes that time is slotted. At the beginning of a time slot t, each link (a, b)'s link-weight W_{ab} is assigned by the maximal flow-weight, that is, the maximum queue length differential among all flows' queues that the two nodes maintain, which is as follows:

$$W_{ab}(t) = \max_{f:(a,b)} \left[U_a^f(t) - U_b^f(t)\right], \qquad (7)$$

where $U_a^f(t)$ denotes the queue length for flow f on node a at time slot t. Recall the characteristics of flows and queues in a WSN as we have mentioned previously; that is, all data packets in network can be seen as belonging to one flow; (7) can therefore be rewritten as follows for simplicity:

$$W_{ab}(t) = U_a(t) - U_b(t). \qquad (8)$$

Packets will be transmitted on link (a, b) if (a, b) is selected by a schedule $\pi(t)$ which is derived from the following optimization problem:

$$\pi(t) = \arg\max_{\pi \in \Gamma} \sum_{(a,b)} W_{ab}(t)\, r_{ab}(t), \qquad (9)$$

where Γ represents the set of all feasible schedules according to given link interference model and r_{ab} represents the link rate of (a, b).

4.2. Motivation. Energy use efficiency is a big issue for backpressure based algorithm to be used in a WSN. In

Table 1: The death time (slot) when the first node dies.

Algorithms	Arrival rates (packets/slot)		
	0.5	1	1.5
BP	247	220	217
Min-hop routing	764	444	392

Table 2: The survival ratio of nodes when the simulation terminates.

Algorithms	Arrival rates (packets/slot)		
	0.5	1	1.5
BP	36%	26%	22%
Min-hop routing	98%	92%	92%

backpressure based transmission scheduling, packets are always forwarded away from network hot-spot (pushed by the so-called backpressure), which is consequently very helpful for balancing the network load and fully utilizing the network capacity. However, such routing and scheduling do not consider whether a routing selection decision in this way leads to routing detours or even loops, which often consumes more energy for packet delivery than shortest paths.

To present a clear understanding regarding this, we next present a simulation based comparison between the classical backpressure algorithm (referred to as BP) and min-hop routing (i.e., each node always chooses a next-hop forwarder from its neighbor nodes which are closer to the sink than itself, which is referred to as Min-hop) in terms of their energy use performance in a WSN. In the simulations, a WSN constituent of 99 sensor nodes and one sink node is used, where the network topology is randomly generated. Link capacity is set to one. The initial energy of each sensor node is assigned to 200 J, and the sink has infinite energy. Sending and receiving a packet cost 1.6 J and 1.0 J, respectively. Each simulation lasts 1000 time slots. Either algorithm's energy use performance was evaluated under different flow arrival rates (i.e., 0.5, 1, and 1.2 packets/slot) and in terms of the following two measures: the death time of the first dead node in the network and the survival ratio of nodes when a simulation comes to the end. From the results in Tables 1 and 2, it is seen that the classical backpressure algorithm performs much worse than the gradient based routing algorithm in terms of energy use efficiency. Under BP, the node survival ratio is extremely low, which reveals the backpressure based algorithms' unsuitability for WSNs. To the best of our knowledge, no work has been done regarding how to improve the energy use performance of back pressure based algorithms in a WSN. In this paper, we take the first step towards this direction. Specifically, we try to answer the following two questions:

(i) How to suppress the use of unnecessarily long routes in backpressure based routing selection.

(ii) How to select nodes with abundant residual energy to undertake forwarding tasks while still preserving backpressure algorithm's throughput-optimality.

In the next, we present the design details of GRAPE and explain how it addresses the above issues.

4.3. Algorithm Design. In GRAPE, besides nodal queue backlog status, the following two new factors (i.e., each node's gradient information and also its residual energy) are introduced into the backpressure based scheduling decision making process. In this paper, the gradient associated with a node is its hop distance to the sink node and the introduction of this factor is to encourage packets to travel along shorter routes. The introduction of nodal residual energy status is to enable backpressure scheduling to be energy aware when selecting next hop nodes. For this purpose, a new link-weight calculation method is presented as follows.

Specifically, in GRAPE, the weight of a link is no longer simply equal to the queue length differential between the two end nodes of the link as shown in (8). Instead, a new link-weight is assigned as follows:

$$W'_{ab}(t) = \Delta U_{a,b} + V_{a,b}, \qquad (10)$$

where $\Delta U_{a,b}$ denotes the queue length differential on link (a, b); that is,

$$\Delta U_{a,b} = U_a(t) - U_b(t). \qquad (11)$$

$V_{a,b}$ is defined as a selecting-bias factor, which is a parameter affecting the probability of activating link (a, b), that is, the possibility of node a choosing its neighbor b as the next-hop forwarder. That is, the smaller $V_{a,b}$ is, the lower the possibility of selecting link (a, b) into final schedule set will be. We use such node selecting-bias factor to discourage packets to be sent to nodes which are not expected to be chosen, for example, nodes with lower residual energy or farther away from the destination than the sender itself. For this purpose, the selecting-bias factor $V_{a,b}$ of link (a, b) is determined based on the receiver b's residual energy status and also nodes a and b's gradients, which is calculated as follows:

$$V_{a,b} = \begin{cases} kG_a, & \text{if } b \text{ is the sink node} \\ k(G_a - G_b) + e^{E_C^b/E_P^b}, & \text{otherwise.} \end{cases} \qquad (12)$$

In (12), G_a represents node a's gradient. E_P^b represents node b's initial energy and E_C^b represents node b's current residual energy. $e^{E_C^b/E_P^b}$ is determined via extensive simulations by comparing with several other options. k is a constant parameter which is also tunable in simulations. Under (12), the selecting-bias is decided by nodes' gradients and candidate next hop receiver's residual energy status. From (10) and (12), it can be seen that, for node a, if its neighbor b has higher residual energy in a's one hop scope, this may result in a higher $\Delta V_{a,b}$ and thus increased probability for it to be chosen as a next hop forwarder into the final schedule. Similarly, the introduction of node gradient reduces the probability of routing loops. By using such routing selection-bias, considerations for nodal energy and path length are introduced into backpressure based scheduling decision making process in a moderate manner. That is, in GRAPE, the network capacity can still be fully utilized, and only a selecting-bias is added during link-weight calculation, where links with lower weights are not forbidden but just discriminated (to some extent) to be chosen. We will further theoretically analyze that transmission scheduling in such a way that will not violate backpressure based algorithm's throughput-optimality in the next subsection.

After determining the link-weight for each link in the network, data packets can then be scheduled to transmit over a link (a, b) if (a, b) is to be activated under a schedule $\pi(t)$ which is derived based on the following optimization problem:

$$\pi(t) = \arg\max_{\pi\in\Gamma} \sum_{(a,b)} W'_{ab}(t)\, r_{ab}(t). \quad (13)$$

In GRAPE, scheduling decisions are made by choosing links whose link-weights can yield a global maximal sum, in which nodes with higher residual energy have higher probabilities to be chosen as relay nodes, as calculated in (10)–(12).

The optimal solution to (13) yields the optimal schedule and its computation needs to be done in a centralized manner and has high computational complexity of at least $O(|V|^3)$ based on which link interference model is used, where $|V|$ represents the number of nodes in the network. To reduce the computational complexity, in [22], Lin and Shroff proposed a distributable Greedy Maximal Matching (GMM) scheme, which works as follows. To compute a schedule (whose initial value is null), add a link (a, b) with the largest weight $W_{ab}(t)$ into the schedule, remove all the links interfering with the link (a, b), and repeat the above link choosing process until no link left. The computational complexity of GMM is $O(|E|\log|E|)$, where $|E|$ represents the number of links in the network.

4.4. GRAPE's Throughput-Optimality. In this section, we provide a theoretical proof regarding the throughput-optimality property of GRAPE.

First, for a WSN, we can substitute (11) and (12) into (10), and accordingly we have

$$W'_{ab}(t) = U_a(t) - U_b(t) + k(G_a - G_b) + e^{E^b_C/E^b_P}. \quad (14)$$

Then, we rewrite (14) in the following form:

$$W'_{ab}(t) = (U_a(t) + kG_a) - \left(U_b(t) + kG_b - e^{E^b_C/E^b_P}\right). \quad (15)$$

Here, if we denote the components kG_a and $kG_b - e^{E^b_C/E^b_P}$ in the right part of (15) as node-specific functions, respectively, and use $Q_{a,s}(t)$ to denote kG_a when node a plays the sending role and use $Q_{b,r}(t)$ to denote $kG_b - e^{E^b_C/E^b_P}$ when b plays the receiving role at time t, we can rewrite (8) as follows:

$$W'_{ab}(t) = (U_a(t) + Q_{a,s}(t)) - (U_b(t) + Q_{b,r}(t)), \quad (16)$$

where it can be seen that, at each time slot t, for $\forall a, b \in V(G)$, $\lim_{a\in V(G)} Q_{a,s}(t) = \lim_{b\in V(G)} Q_{b,r}(t) = O(|V|)$ can always hold (recall that G_a denotes node gradient and is restricted by network size). Consequently, GRAPE actually shares the same scheduling pattern as the EDR in [3] and therefore the same Lyapunov function $L(\mathbf{U}) = \sum_a U_a^2$ used in [3, 4] can be used for proving the throughput-optimality of GRAPE. For the integrality of the paper, we provide them as follows.

For each forwarding queue kept at sensors in the network, its dynamics meets the following expression:

$$U_a(t+1) \le \max\left\{U_a(t) - \sum_n \mu_{an}(t), 0\right\} + \sum_m \mu_{ma}(t) + I_a(t), \quad (17)$$

where $I_a(t)$ denotes the external arrival rates. Consider the Lyapunov function:

$$L(\mathbf{U}) = \sum_a U_a^2, \quad (18)$$

where $\mathbf{U}(t) = \{U_a(t)\}_{a\in V(G)}$. The Lyapunov drift $\Delta(\mathbf{U}(t))$ can then be derived as follows:

$$\Delta(\mathbf{U}(t)) \le \mathbb{E}\left[\sum_a U_a(t+1)^2 - \sum_a U_a(t)^2 \mid \mathbf{U}(t)\right]. \quad (19)$$

Based on the fact that $(\max(U-b, 0) + A)^2 \le U^2 + A^2 + b^2 + 2U(A - b)$, we can rewrite (16) as follows:

$$\begin{aligned}\Delta(\mathbf{U}(t)) \le \mathbb{E}\Bigg[&\sum_a \Bigg(U_a(t)^2 + \left(\sum_m \mu_{ma}(t) + I_a(t)\right)^2 \\ &+ \sum_n \mu_{an}(t)^2 \\ &+ 2U_a(t)\left(\sum_m \mu_{ma}(t) + I_a(t) - \sum_n \mu_{an}(t)\right)\Bigg) \\ &- \sum_a U_a(t)^2 \mid \mathbf{U}(t)\Bigg].\end{aligned} \quad (20)$$

Since there always exists a finite constant $\mathbb{B}$ such that $\mathbb{B} \ge \mathbb{E}[\sum_a \sum_n \mu_{an}(t)^2 + \sum_a(\sum_m \mu_{ma}(t) + I_a(t))^2 \mid \mathbf{U}(t)]$, we have

$$\begin{aligned}\Delta(\mathbf{U}(t)) &\le \mathbb{B} \\ &+ 2\mathbb{E}\left[\sum_a U_a(t)\left(\sum_m \mu_{ma}(t) - \sum_n \mu_{an}(t)\right)\right. \\ &\left.+ \sum_a U_a(t) I_a(t) \mid \mathbf{U}(t)\right].\end{aligned} \quad (21)$$

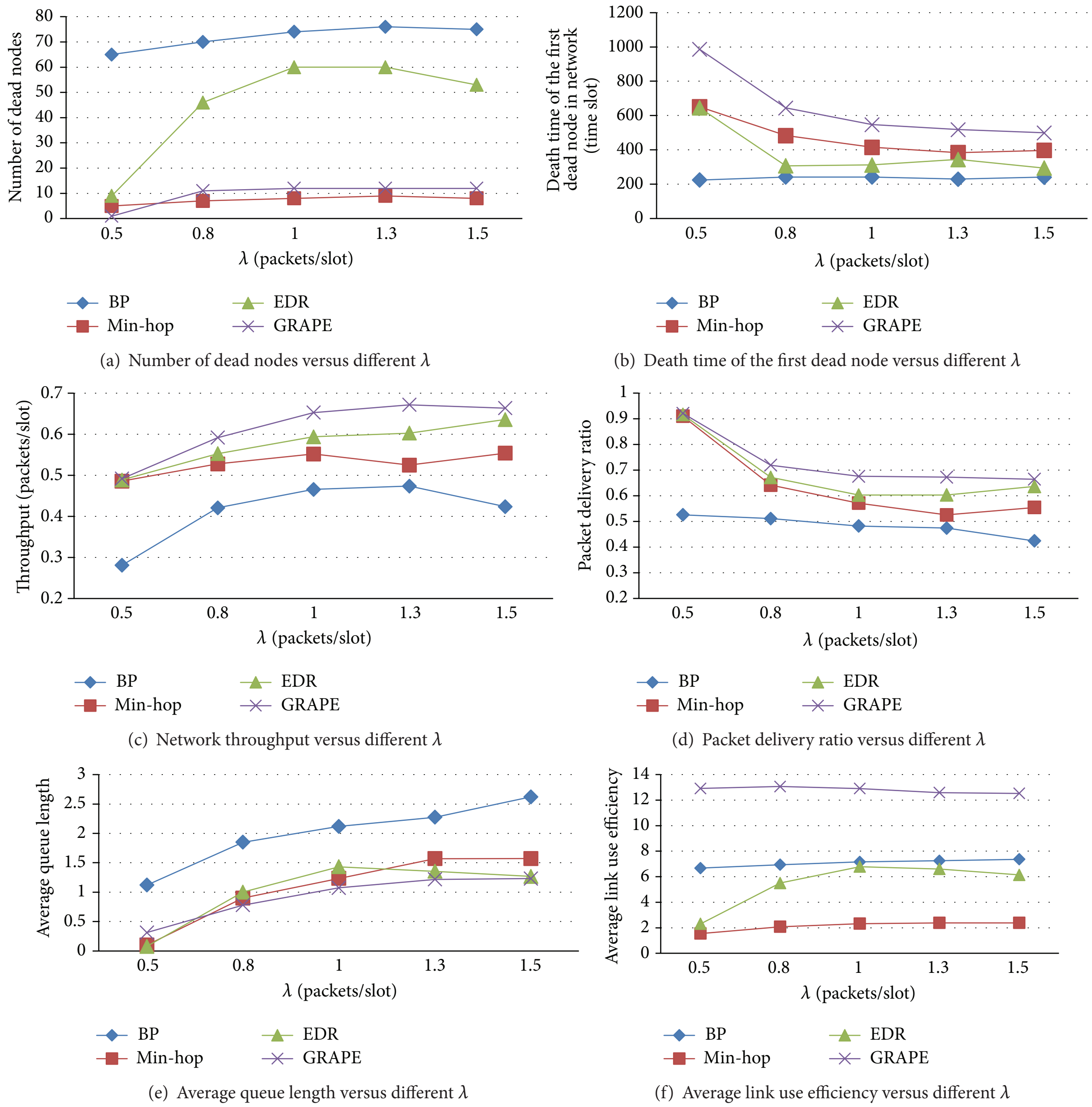
The arrival rates are assumed to be within the capacity region; as a result, there always exists a constant $\epsilon > 0$ such that $\mathbb{E}[\sum_a(\sum_m \mu_{ma}(t) - \sum_n \mu_{an}(t)) \mid \mathbf{U}(t)] \le -(I_a(t) + \epsilon)$. By substituting it to (18), we can have

$$\Delta(\mathbf{U}(t)) \le \mathbb{B} - 2\sum_a U_a(t)\,\epsilon. \quad (22)$$

Thus, we have

$$\lim_{t\to\infty}\sup \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_a U_a(t) < \frac{\mathbb{B}}{\epsilon}, \quad (23)$$

which means, just like the algorithms proposed in [3, 4], the transmission scheduling by GRAPE in this paper can always make queues in the network to be bounded when the network arrival rates are located within the network capacity; that is to say, GRAPE is throughput optimal.



(a) Number of dead nodes versus different λ

(b) Death time of the first dead node versus different λ

(c) Network throughput versus different λ

(d) Packet delivery ratio versus different λ

(e) Average queue length versus different λ

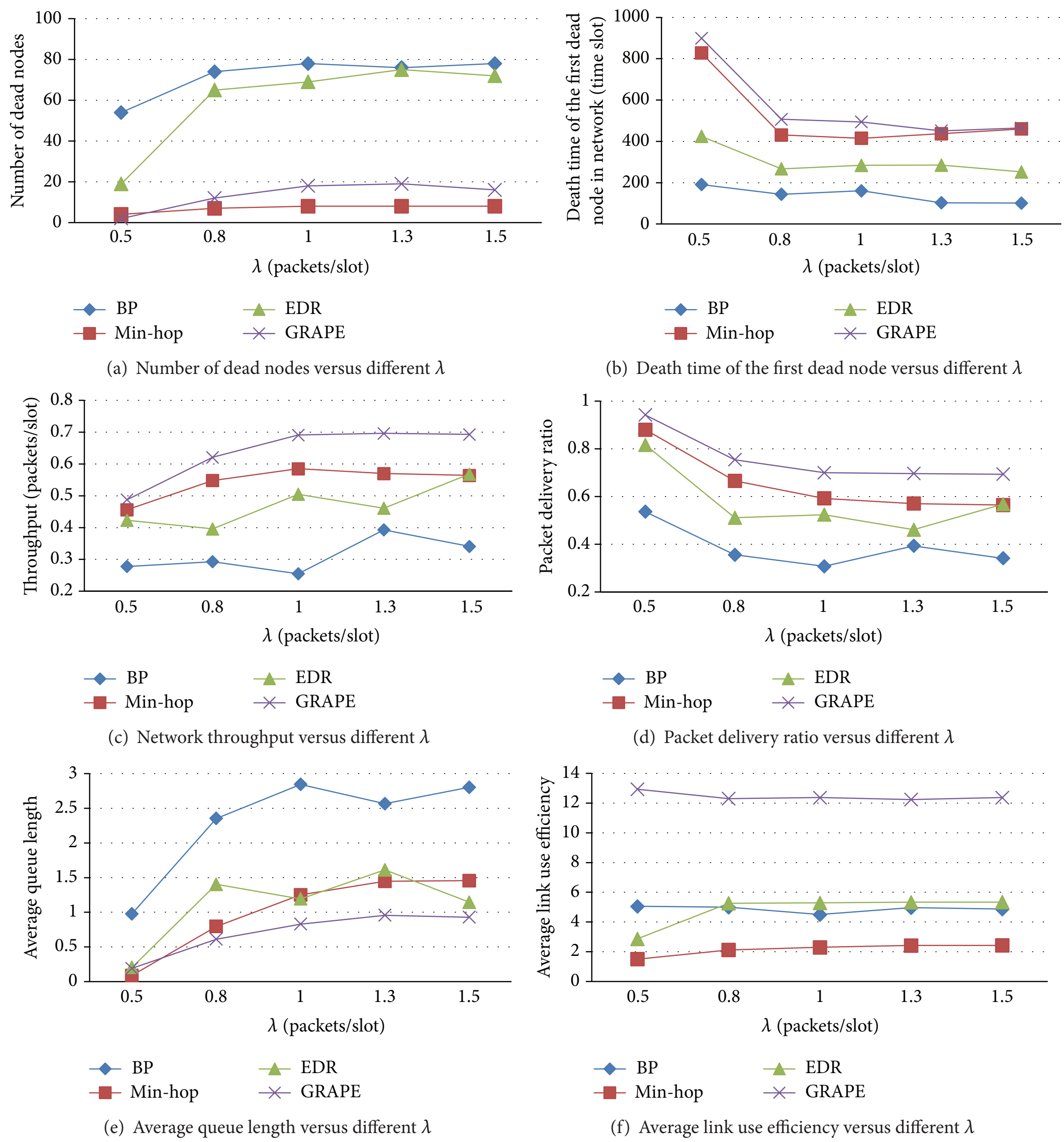
(f) Average link use efficiency versus different λ

Figure 1: Performance comparison of various algorithms on a randomly generated 100-node topology when link capacity equals one.

5. Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of GRAPE by comparing it with several other existing algorithms including the classical backpressure algorithm (referred to as BP) [1], the enhanced dynamic back-pressure routing algorithm (EDR) in [3], and the min-hop routing algorithm (referred to as Min-hop). Next, we will first introduce our simulation settings and then present the simulation results.

In the simulations, we generated a random topology with 100 nodes located within a 500×500 square area. The communication radius of each node is 100. In the network, a randomly chosen node acts as the sink node in the network and all other nodes are sensor nodes. Packets can be injected into the network via any sensor node where packet arrival follows a Poisson process with arrival rate λ. In the simulations, we will present comparison results under different packet arrival rates λ, where λ varies with the following values: 0.5, 0.8, 1.0, 1.3, and 1.5. To estimate different algorithms' energy use performance, each sensor node is assigned by 200 J initial energy, and the sink is assigned by infinite energy. Sending and receiving a packet cost 1.6 J and 1.0 J to a node, respectively. Each simulation lasts for 1000 slots. We use two metrics to exhibit the energy use efficiency between different algorithms, that is, the death



(a) Number of dead nodes versus different λ

(b) Death time of the first dead node versus different λ

(c) Network throughput versus different λ

(d) Packet delivery ratio versus different λ

(e) Average queue length versus different λ

(f) Average link use efficiency versus different λ

Figure 2: Performance comparison of various algorithms on a randomly generated 100-node topology when link capacity equals five.

time of the first dead node and number of dead nodes, which are widely used metrics for measuring network life time and nodal energy usage efficiency in existing work.

Figure 1 shows the simulation results when network link capacity is set to one. Figures 1(a) and 1(b) compare the energy use performance by different algorithms, where Figure 1(a) shows the numbers of dead nodes under different algorithms when the simulation terminates and Figure 1(b) compares the death times of the first dead nodes by different algorithms, respectively. From these two subfigures, it is seen that GRAPE outperforms other backpressure based algorithms (i.e., BP and EDR) in terms of energy use efficiency, which validates the high energy use efficiency of our new link-weight calculation method used in GRAPE. However, Min-hop can still outperform GRAPE in some cases. The reason is that Min-hop restricts packets to be transmitted from nodes with higher gradients to those with lower gradients. It forbids the use of any longer alternate paths other than shortest paths. Thus, it can have higher energy use performance than backpressure based algorithms including GRAPE, since backpressure based algorithms leverage alternate routes to fully utilize the network capacity, which is a key feature of backpressure based scheduling and can cause more energy consumption. Figures 1(c) and 1(d) demonstrate the network throughput and packet delivery ratio performance by different algorithms. It is clearly seen that GRAPE outperforms Min-hop in terms of these two measures especially as the traffic arrival rate is increasing. This is due to adaptive backpressure based routing's capability in fully utilizing alternate routes. Figure 1(e) compares the average queue length performance under different algorithms. It can also be seen that, due to better throughput performance of GRAPE, the average queue length under GRAPE is lower than the other three algorithms as traffic arrival rate increasing. Here, note that when flow arrival rate is 0.5, the average queue length by GRAPE is higher than Min-hop. The reason is that backpressure based algorithms always need some time to form queue based gradient in the network to act as back pressure for pushing packets to go. As a result, when flow rate is low, packets may need to stay at a node and wait for a longer time than that when arrival rate is high, which result in longer average queue length. This phenomenon is called the slow-startup problem of backpressure based scheduling in [4]. For more details please refer to [4]. Furthermore, we compared the link use efficiency by each algorithm, which is defined by the number of links being activated in each time slot. Average link use efficiency equals to the average number of links being chosen into the schedule set generated by an algorithm per time slot, which is an important metric for estimating a backpressure based algorithm's scheduling efficiency. As shown in Figure 1(f), GRAPE has higher link use efficiency than other algorithms due to its higher energy use efficiency (since in our simulation, nodes that had exhausted their energy will no longer participate in any transmissions). Furthermore, it should be noted that Min-hop always activates much less links per time slot than backpressure based algorithms. This is because Min-hop forces packets to be transmitted along shortest paths. As a result, in some hot-spots, few links can be activated due to contentions in medium accessing opportunities. This is helpful for saving energy but cause reduced network capacity.

Figure 2 compares the performance of different algorithms when network link capacity was set to five. In Figure 2, it is again seen that GRAPE outperforms the BP and EDR in terms of energy use efficiency, throughput, packet delivery ratio, average queue length, and link use efficiency. Furthermore, it has comparable energy use performance to Min-hop as shown in Figures 2(a) and 2(b).

6. Conclusion

Energy use performance is always a big design concern for backpressure based routing and scheduling to be useful in a resource-constrained wireless sensor network. In this paper, we proposed GRAPE, a gradient-assisted energy-efficient backpressure scheduling algorithm for WSNs. In GRAPE, besides queue backlog differentials, gradient information and nodal residual energy are also jointly considered into the transmission scheduling decision making process and accordingly a new link-weight calculation method was designed, according to which packets are encouraged to be forwarded to nodes with more residual energy and via shorter paths. We present the detailed design description of GRAPE and further theoretically prove its throughput-optimality. Extensive simulations results show that GRAPE significantly outperforms existing algorithms in terms of energy use efficiency, packet delivery ratio, and throughput.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61501125, 61502018, 61531006, and 61372105 and Natural Sciences and Engineering Research Council (NSERC) of Canada (Discovery Grant 293264-12 and Strategic Project Grant STPGP 397491-10).

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[2] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: the backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, pp. 279–290, IEEE, April 2010.

[3] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[4] Z. Jiao, B. Zhang, W. Gong, and H. Mouftah, "Virtual queue based back-pressure scheduling algorithm for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 35, 2015.

[5] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 841–854, 2011.

[6] L. A. Maglaras and D. Katsaros, "Layered backpressure scheduling for delay reduction in ad hoc networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '11)*, pp. 1–9, Lucca, Italy, June 2011.

[7] E. Athanasopoulou, L. X. Bui, T. Ji, R. Srikant, and A. Stolyar, "Back-pressure-based packet-by-packet adaptive routing in communication networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 244–257, 2013.

[8] L. X. Bui, R. Srikant, and A. L. Stolyar, "A novel architecture for reduction of delay and queueing structure complexity in the backpressure algorithm," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1597–1609, 2011.

[9] L. Ying, R. Srikant, D. Towsley, and S. Liu, "Cluster-based back-pressure routing algorithm," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1773–1786, 2011.

[10] Z. Jiao, Z. Yao, B. Zhang, and C. Li, "A distributed gradient-assisted anycast-based backpressure framework for wireless sensor networks," in *Proceedings of the 1st IEEE International*

Conference on Communications (ICC '14), pp. 2809–2814, Sydney, Australia, June 2014.

[11] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multi-hop wireless networks," in *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 2579–2587, IEEE, Shanghai, China, April 2011.

[12] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. J. Neely, "Backpressure with adaptive redundancy (BWAR)," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 2300–2308, March 2012.

[13] A. Dvir and A. V. Vasilakos, "Backpressure-based routing protocol for DTNs," in *Proceedings of the 7th International Conference on Autonomic Computing (SIGCOMM '10)*, pp. 405–406, New Delhi, India, September 2010.

[14] H. Seferoglu and E. Modiano, "Diff-Max: separation of routing and scheduling in backpressure-based wireless networks," in *Proceedings of the IEEE (INFOCOM '13)*, pp. 1555–1563, IEEE, Turin, Italy, April 2013.

[15] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, "DiffQ: practical differential backlog congestion control for wireless networks," in *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM '09)*, pp. 262–270, IEEE, Rio de Janeiro, Brazil, April 2009.

[16] B. Radunović, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: balancing TCP over multiple paths in wireless mesh network," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 247–258, ACM, San Francisco, Calif, USA, September 2008.

[17] R. Laufer, T. Salonidis, H. Lundgren, and P. L. Guyadec, "XPRESS: a cross-layer backpressure architecture for wireless multi-hop networks," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom'11)*, pp. 49–60, ACM, September 2011.

[18] Z. Jiao, Z. Yao, B. Zhang, and C. Li, "NBP: an efficient network-coding based backpressure algorithm," in *Proceedings of the IEEE International Conference on Communications (ICC '13)*, pp. 1625–1629, Budapest, Hungary, June 2013.

[19] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool, 2010.

[20] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89–103, 2005.

[21] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1–14, Berkeley, Calif, USA, November 2009.

[22] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, 2006.