2018 IEEE 15th International Conference on Mobile Ad-hoc and Sensor Systems

Multi-hop Deflection Routing Algorithm based on Q-Learning for Energy-harvesting Nanonetworks

Chao-Chao Wang*, Qing Xia†, Xin-Wei Yao*, Wan-Liang Wang*, Josep Miquel Jornet†
* College of Computer Science and Technology
Zhejiang University of Technology, Hangzhou 310023, China
Email: {ccwang, xwyao, zjutwwl}@zjut.edu.cn
† Department of Electrical Engineering, University at Buffalo
The State University of New York, Buffalo, New York 14260, USA
Email: {qingxia, jmjornet}@buffalo.edu

***Abstract*—Nanonetworks composed by communicating nano-devices enable new applications in the consumer, biomedical, and environmental fields. Three main characteristics introduce strict requirements for routing protocols design for nanonetworks, namely, short transmission range at Terahertz (THz) frequency (0.1-10 THz), fluctuations in the energy of nano-nodes due to the energy harvesting processes and very limited memory/buffer size of nano-nodes. In this paper, a multi-hop deflection routing algorithm based on Q-learning for energy-harvesting nanonetworks (MDRQEN) is proposed to guarantee the network energy efficiency, while ensuring a low packet loss probability. First, a deflection table is introduced to deflect the packets when the next hop nano-nodes are unavailable due to energy or memory/buffer constraints. Then, a Q-learning scheme is proposed to update the routing table and deflection table by utilizing the reward information contained in the forwarded packet from the previous nano-node. In the Q-learning update scheme, packet deflection ratio, packet loss ratio, packet hop count and node energy status of nano-nodes are taken into consideration. As numerically shown through extensive simulations in Network Simulator 3 (NS-3), the proposed MDRQEN algorithm can achieve a better packet delivery ratio and energy efficiency than random routing algorithm, flooding routing algorithm and the MDRQEN algorithm without the Q-learning update scheme.**

I. Introduction

Nanotechnology enables the development of nanonetworks, i.e., assemblies of communicating nano-devices or nano-nodes, whose size is in the order of a few cubic micrometers. The applications of nanonetworks range form biomedical, environmental and military fields, such as intra-body health monitoring system, water pollution control and bio-hazard defenses [1], [2]. Due to the extremely small size of nano-nodes, and therefore nano-antennas, high frequency bands are used for wireless communication in electromagnetic nanonetworks. Recently, the progress in graphene-based electronics has enabled nano-devices communication in the Terahertz (THz) band (0.1-10 THz) [3].

In the last few years, many contributions have been made in terms of nano-node design [4]–[6], as well as physical [7]–[9] and link layer design [10]–[12]. However, there are only a few studies focused on information routing, which is an important issue for the efficient delivery of packets across the nanonetworks.

Compared to the traditional wireless sensor networks (WSNs), the peculiarities of nanonetworks introduce new challenges in efficient routing protocol design. Firstly, as a result of the very high propagation loss at THz frequency and the limited power of THz signals, the transmission range of nano-nodes is drastically limited. This fact indicates multi-hop routing algorithms are needed to allow nano-nodes transmit packets from sources to destinations. Secondly, due to the limited size of nano-nodes, batteries need to be scaled down to a few hundred cubic nanometers, which limits the energy capacity. Hence, energy harvesting nano-systems are needed, such as piezoelectric nano-generators [13], [14]. The resulting fluctuations in the energy of nano-nodes make the routing paths unstable. Therefore, the routing algorithms for nanonetworks should be energy efficient and able to adapt to the dynamic energy status of nano-nodes. Thirdly, the extremely small size of nano-nodes also limits the onboard memory/buffer size. Because of this, traditional store and forward packets routing policy might not work in nanonetworks. During the time a nano-node is looking up for the best route for a packet in its buffer, it can not process the other packets which are effectively lost. Because of all the reasons, routing algorithms in WSNs can not be directly applied to nanonetworks.

Recently, some routing algorithms for nanonetworks are proposed to improve the energy efficiency. Most of the them are flood-based, where nano-nodes send packets to all the nano-nodes within their transmission range, and thus, the algorithms try to reduce the energy consumption of nano-nodes by restricting the flooding area. In [15], the authors proposed a joint coordinate and routing system (CORONA) for uniformly distributed nanonetworks in a rectangular area. Four anchors are placed at the area vertexes sending coordinates (in terms of hop count) to all the nano-nodes. When a nano-node wants to send packets to another node, the intermediate nodes whose coordinates are between the two nodes flood the packet to the destination. This algorithm can improve the energy efficiency in nanonetworks, but has high restrictions with the topology of the network. In [16], a deployment routing system (DEROUS) is proposed for the centralized nanonetworks. In the DEROUS system, a beacon node is deployed in the center of the nanonetwork, and all the other nodes set the hop

2155-6814/18/$31.00 ©2018 IEEE
DOI 10.1109/MASS.2018.00059

counts driven by distances from the central beacon node. The flooding transmissions are restricted between the nano-nodes whose hop counts from the central beacon are between the transmitting nano-node and receiving nano-node. An energy efficient multi-hop routing protocol (EEMR) is proposed in [17], which narrows the next hop candidates nodes area by controlling the direction of multi-hop forwarding. However, on one hand, the above routing algorithms are flood-based, which indicates that they still need to cost extra energy to deliver packets without routing paths, and thus, results in a low energy efficiency. On the other hand, the lack of memory/buffer is not considered in these routing algorithms, which has significant influences on the energy efficiency and packet loss probability of nanonetworks.

The problem caused by memory/buffer also occurs in buffer-less optical burst switching (OBS) networks [18]. In buffer-less OBS networks, optical bursts can be lost when wavelength contention occurs. This kind of contention appears when more than two optical bursts try to use the same output port to the destinations, on the same wavelength, at the same time. Hence, deflection routing algorithm is proposed as one of the approaches to reduce the burst loss probability. In the deflection routing algorithm, only one of the bursts is routed to the primary output port of a node, where the others are deflected to other alternative output ports when contention occurs. This kind of contention also happens in nanonetworks, when the next hop nano-node in the routing table consumes all its energy or is busy communicating with other nano-nodes. Therefore, nanonetworks can also adopt deflection routing algorithm to reduce the packet loss probability and improve the energy efficiency.

Furthermore, due to the dynamic of traffic loads in the buffer-less OBS networks, the method of choosing the deflected node fixedly or randomly may not suit for the buffer-less OBS networks. In this direction, deflection routing algorithms strengthened by Q-learning algorithm [19] have been studied recently [20], [21]. In such algorithms, a learning agent is installed in each node to interact with its environment by making decisions and receiving rewards. Whenever a nano-node takes an action to deflect a burst to an alternative neighbor, the learning agent receives one or several feedback rewards, e.g., the hop count to the destination and the packet loss probability. Then the agent updates the priorities of the alternative neighbors for the future deflection.

However, all the Q-learning deflection routing algorithms in buffer-less OBS networks can update the deflection tables by the feedback information only under the assumption that each node has the shortest path routing table. Nevertheless, in nanonetworks, it is hard for the nano-node to obtain a shortest path routing table initially. Furthermore, all the Q-learning deflection routing algorithms in buffer-less OBS networks do not consider the energy consumption of nodes which is quite important for nanonetworks. In nanonetworks, nano-nodes may become unavailable before it harvests enough energy to recieve/forward packets, which results in packet loss if without deflection. Hence, both the dynamic traffic loads and energy statuses of nano-nodes should be considered in the deflection routing algorithm for nanonetworks.

In this paper, a multi-hop deflection routing algorithm based on Q-learning (MDRQEN) is proposed to reduce the packet loss probability and to improve the energy efficiency. In the proposed MDRQEN algorithm, a deflection table is introduced to deflect the packets when the next hop nano-node in the routing table is unavailable due to energy or memory/buffer constraint. Moreover, a Q-learning update scheme is proposed to update the routing table and deflection table by utilizing the reward information (considering packet deflect ratio, packet loss ratio, packet hop count and node energy status) contained in the forwarded packet from previous nano-node. Extensive simulations in Network Simulator 3 (NS-3) have been conducted to compare the performance of the MDRQEN algorithm with random routing algorithm, flooding routing algorithm and proposed MDRQEN algorithm without the Q-learning update scheme in terms of packet delivery ratio, number of delivery packet and packet average hop count. From the results, we conclude that our proposed MDRQEN algorithm can achieve a better performance than the other three routing algorithms.

The remainder of the paper is organized as follows. The details of the proposed deflection routing based on Q-learning algorithm is introduced in Sec. II. In Sec. III, we presents the comparison of simulation results among the proposed MDRQEN algorithm with some other routing algorithms in terms of packet delivery ratio, number of delivered packets and average hop count. Finally, we conclude the paper in Sec. IV.

II. Deflection Routing Algorithm based on the Q-learning Algorithm

In this section, the details of the proposed multi-hop deflection routing algorithm based on Q-learning (MDRQEN) are presented. First, we introduce the routing table and deflection table structures, as well as the Q-learning scheme which is utilized to update them. Then, we present the operation of the proposed MDRQEN algorithm in detail.

The main objective of our algorithm is to find an optimal next hop nano-node to transmit the packet. In the proposed MDRQEN algorithm, a deflection table is introduced to deflect the packets when the route entry in the routing table is invalid due to the contentions or energy/buffer issues. Both the routing table and deflection table are built up by the forwarded packets from previous nano-node and updated based on the proposed Q-learning scheme.

A. Q-learning Scheme for Tables Update

1) Table Structures: An arbitrary nanonetwork is shown in Fig. 1. We consider each node maintains a routing table and a deflection table, and all the nano-nodes implement the proposed MDRQEN algorithm. In the routing table, there is only one route entry for a certain destination. If a nano-node needs to transmit packets to the destination, it firstly looks up the routing table for the next hop nano-node. The routing table adopts the following fields with each route entry:

- Destination nano-node ID

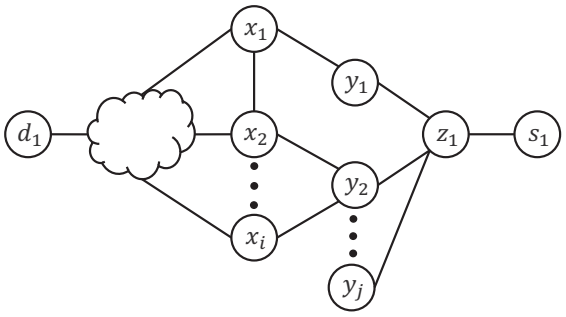

Fig. 1. An arbitrary nanonetwork

- Next hop nano-node ID
- Q-value to the destination via next hop nano-node
- Recovery rate to the destination via next hop nano-node
- Hop count to the destination
- Time when this route is updated
- Route valid flag (updated when receive an ACKnowledgement (ACK) or a Negative ACKnowledgement (NACK) packet or do not receive the ACK packet)
- Lifetime

where Q-value represents the weight of the corresponding route, bigger Q-value means the route is worse. Recovery rate is applied to adapt to the change of the energy statuses of nano-nodes and the traffic load of network by time. For example, the energy of nano-node recovers from harvesting energy from environment, the traffic load becomes bigger when nano-nodes send message more frequently, which would influence the Q-value of routes. Initially, the routing tables in nano-nodes could be empty.

Due to the extremely small size, nano-nodes have very limited energy and memory/buffer to transmit packets. Hence, route entries in the routing table could be invalid when the following situations happen: (i) the next hop nano-node consumes all its energy; (ii) the next hop nano-node is busy communicating with another nano-node; (iii) error occurs during the transmission. Therefore, a deflection table associated with nodes' neighboring is introduced to deflect the packet to another nano-node if the route entry in the routing table is invalid, this is then to prevent the packets being dropped due to the limited energy or memory. In a deflection table one nano-node may have several route entries for one destination, e.g., as shown in Fig 1, z_1 can transmit packets to d_1 through y_1 or y_2. In a deflection table, each entry is indexed by destination and neighbor. Considering an arbitrary nano-node z_1 as example, each deflection route entry contains the following fields:

- $Q_{z_1}(d, y)$ - Q-value from node z_1 to destination d via neighboring node y
- $R_{z_1}(d, y)$ - Recovery rate for the Q-value from node z_1 to destination d via neighboring node y
- $H_{z_1}(d, y)$ - Hop Count from node z_1 to destination d via neighboring node y
- $T_{z_1}(d, y)$ - Time when this route entry is updated

2) Deflection Strategy: Consider that z_1 receives a packet from s_1 to d_1. Firstly, it looks up the routing table, if the route entry is invalid, z_1 deflects the packet to its neighbor y_m determined by

$$m = \arg\min\left(Q_{z_1}(d_1, y_j) + \Delta t R_{z_1}(d_1, y_j)\right), \quad (1)$$

where arg min is to obtain the index of the next hop nano-node with minimum Q-value. y_j is the group of neighboring nano-nodes of z_1 except the one has been discarded. $R_{z_1}(d_1, y_j)$ is the recovery rate for the Q-value from node z_1 to destination d via neighboring node y_j. Δt is the time duration between the time when the table entry is updated and current time t_c which can be obtained by:

$$\Delta t = t_c - T_{z_1}(d_1, y_j), \quad (2)$$

3) Q-learning Update Scheme: The deflection table is updated upon the receive of a forwarded packet. Continuing with the same example, we consider that z_1 receives a packet from d_1 forwarded by y_1 through x_1. The header of the forwarded packet contains a reward which can be expressed as follows:

$$\begin{aligned} r_{y_1}(d_1, x_1) =& Q_{y_1}(d_1, x_1) \cdot (H_{y_1}(d_1, x_1) + 1) \cdot \\ & P^{y_1}_{deflected} \cdot P^{y_1}_{loss} \cdot C^{y_1}_{energy}, \end{aligned} \quad (3)$$

where $Q_{y_1}(d_1, x_1)$ and $H_{y_1}(d_1, x_1)$ are the Q-value and hop count from y_1 to d_1 through x_1, respectively. $P^{y_1}_{deflected}$ is the deflection probability of y_1 which can be obtained as

$$P^{y_1}_{deflected} = \frac{N^{y_1}_{def}}{N^{y_1}_{send}}, \quad (4)$$

where $N^{y_1}_{def}$ is the number of times that deflection occurs in y_1, $N^{y_1}_{send}$ is the number of packets transmitted by y_1, including both the generated and forwarded traffics. $P^{y_1}_{loss}$ is the drop probability of y_1, given by

$$P^{y_1}_{loss} = \frac{N^{y_1}_{loss}}{N^{y_1}_{send}}, \quad (5)$$

where $N^{y_1}_{loss}$ is the number of lost packets. $C^{y_1}_{energy}$ is the percentage of consumed energy of y_1, which can be obtained as

$$C^{y_1}_{energy} = \frac{E_{\max} - E^{y_1}_c}{E_{\max}} \times 100\%, \quad (6)$$

where $E_{\max}$ is the maximum energy capacatiy of a nano-node, $E^{y_1}_c$ is the current energy capacity of a y_1.

When z_1 receives the forwarded packet from y_1, it extracts the reward r_{y_1} from the packet and updates its corresponding deflection route entry to the destination as follows:

$$\begin{aligned} Q_{z_1}(d_1, y_1) =& Q_{z_1}(d_1, y_1) + \\ & \alpha\left(\frac{r_{y_1}(d_1, x_1)}{H^r_{z_1}(d_1)} - Q_{z_1}(d_1, y_1)\right), \end{aligned} \quad (7)$$

where $H^r_{z_1}(d_1)$ is the hop count to destination d_1 from the routing table, $\alpha\,(0 < \alpha \leq 1)$ is the learning rate, which decides

Algorithm 1 Tables Update operations of MDRQEN algorithm

Input: Receive a packet

Update the routing and deflection tables :

if (Route entry to source does not exist) **then**
  if (Routing table or deflection table is full) **then**
    Delete the oldest route entry;
    Add the route to the routing table and deflection table;
  else
    Add the route to the routing table and deflection table;
  end if
else
  Update the routing table and deflection table by the Q-learning scheme;
end if

how much the nano-node learn from the reward. Then, the recovery rate $R_{z_1}(d_1, y_1)$ will be updated as follows:

$$R_{z_1}(d_1, y_1) = \begin{cases} R_{z_1}(d_1, y_1) + \beta \frac{\phi}{\Delta t_{y_1}}, & \phi < 0 \\ \gamma R_{z_1}(d_1, y_1), & \phi \geq 0, \end{cases} \tag{8}$$

where $\beta\,(0 < \beta \leq 1)$ and $\gamma\,(0 < \gamma \leq 1)$ are the recovery and decay coefficient respectively, and jointly decide the value of recovery rate. $\phi = \frac{r_{y_1}(d_1, y_1)}{H^r_{z_1}(d_1)} - Q_{z_1}(d_1, y_1)$. Δt_{y_1} is the time duration between $T_{z_1}(d_1, y_1)$ and current time t_c which can be expressed as

$$\Delta t_{y_1} = t_c - T_{z_1}(d_1, y_1). \tag{9}$$

Furthermore, the hop count $H_{z_1}(d_1, y_1)$ is updated by the hop count recorded in the header of the packet. At last, the time record is updated with current time as:

$$T_{z1}(d_1, y_1) = t_c. \tag{10}$$

After updating the deflection table, the route entry which has the same destination in the routing table is updated by comparing $Q^r_{z_1} + \Delta t_{y_1} R^r_{z_1}$ and $Q_{z_1}(d_1, y_1) + \Delta t_{y_1} R_{z_1}(d_1, y_1)$, where $Q^r_{z_1}$ and $R^r_{z_1}$ are the Q-value and recovery rate of the corresponding route entry in the routing table. If the previous is larger, the route entry in the routing table will be replaced by the deflection route entry. Otherwise, the routing table remains unchange.

B. The Operations of MDRQEN Algorithm

Before sending the packet, the network layer adds a header composed by: source nano-node ID, destination nano-node ID, next hop nano-node ID, update information and Time To Live (TTL). The destination nano-node ID and next hop nano-node ID define the final and next nano-nodes which the packet will be sent to, respectively. When the next hop nano-node receives the packet, it checks whether the packet is for it or not, if yes, it will run the proposed MDRQEN algorithm to forward the packet, otherwise, the nano-node will drop the packet.

Algorithm 2 Forward operations of MDRQEN algorithm

Input: Receive a packet

Select the next hop nano-node :

if (The destination is me) **then**
  Process the packet;
else
  if (Energy is not enough to forward the packet) **then**
    Drop the packet;
    Send NACK back;
  else
    Send ACK back;
    Look up the routing table for next hop nano-node;
    if (Route entry in the routing table to the destination is available) **then**
      Update the reward information in the packet;
      Forward the packet to the next hop nano-node;
    else
      Look up the deflection routing table to select the optimal deflection nano-node by the deflection strategy;
      if (Deflection nano-node to the destination exists) **then**
        Update the reward information in the packet;
        Forward the packet to the next hop nano-node;
      else
        if (Neighboring nano-nodes exist) **then**
          Choose a neighboring nano-node randomly as the next hop;
          Update the reward information in the packet;
          Forward the packet to the next hop nano-node;
        else
          Drop the packet;
        end if
      end if
    end if
  end if
end if

Wait for the feedback :

if (No feedback or the feedback is NACK) **then**
  Drop the packet;
  Set the corresponding route entry invalid;
end if

The operations of the proposed MDRQEN algorithm are presented in Algorithms 1 and 2 in detail. Algorithm 1 describes the details of how to update the routing table and deflection routing by the Q-learning scheme. In the MDRQEN algorithm, nano-node uses the forwarded packet to update the routing table and deflection table. When a nano-node receives a packet from another nano-node, it extracts information from the header of the packet. If the route entry to the source nano-node does not exist, the nano-node adds new route entry to the routing table and deflection table. When the routing table or deflection table is full, the oldest entry is deleted. Algorithm 2

TABLE I
PARAMETERS USED IN THE SIMULATIONS

Parameters	Value
Simulation duration	500 s
Density of nano-nodes	[8000 - 16000] nodes/m^2
Packet send request interval	[1 - 9]s
Packet Time To Live	50 hops
Pulse duration	100 fs
Pulse Interarrival Tme	10 ps
Transmission range of nano-nodes	[0.015 - 0.035] m
Learning rate α	0.1
Recovery rate β	0.1
Decay rate γ	0.9
Maximum energy capacity of a nano-node	300 units
Average Energy harvesting speed	[15 - 35] unit/s
Energy consumption of sending a packet	20 units/s
Energy consumption of receiving a packet	10 uints/s
Energy consumption of sending an ACK/NACK	2 units/s
Energy consumption of receiving an ACK/NACK	1 unit/s

expresses the details of how to forward the packet to the next hop nano-node. When a nano-node receives a packet from the others, it checks whether the packet is for itself or not. The nano-nodes forwards the packet only if the energy is enough to forward the packet and receive the feedback. If all the route entries in the routing table and deflection table are invalid or do not exist, it will choose a neighboring nano-node randomly to forward the packet. After receiving an ACK from the next hop nano-node, the process ends. Otherwise, the nano-node drop the packet and set this route entry invalid. If nano-nodes generate packets to be transmitted, the process starts from checking the energy status.

III. Simulations Results

In this section, extensive simulations are conducted to evaluate the performance of the proposed MDRQEN algorithm. Firstly, in Section III-A, we define several main performance metrics of nanonetworks, which include packet delivery ratio, number of delivered packet and packet average hop count. In Section III-B, the simulation platform and parameters setting are summarized. From Section III-C to III-E, simulations are presented by comparing proposed MDRQEN algorithm with random routing, flooding routing and the MDRQEN algorithm without the Q-learning update scheme.

A. Target Performance Metrics

The performance of different routing algorithm is measured in terms of packet delivery ratio, number of delivered packet and packet average hop count which are main performance metrics in nanonetworks. Packet delivery ratio can be obtained as $\frac{N_{delivered}}{N_{generated}}$, where $N_{delivered}$ is the total number of delivered packet, $N_{generated}$ is the total number of generated packet by the nano-nodes. Higher packet delivery ratio indicates a better performance of routing algorithm.

During the comparisons of different routing algorithm, the simulation duration time and energy harvesting speed are identical for all the routing algorithms. Hence, the number of delivered packet $N_{delivered}$ can reflect the energy efficiency and throughput of nanonetworks. Larger number of delivered packet indicates higher energy efficiency and high throughput.

The average packet hop count is used to evaluate the delay of nanonetworks. Larger average packets hop count indicate longer delay.

B. Simulation Platform

To evaluate the performance of all the routing algorithms, we develop a nanonetwork environment in NS-3 to implemented the algorithms. NS-3 is a discrete-event network simulator for Internet systems and is openly available for research and development. In the developed nanonetwork environment, all the nano-nodes have no route entry initially, and are equipped with a sensing unit which can sense the surrounding environment and can trigger sending packet requests in the case of sufficient energy. Consider that the nano-buffer in nano-nodes can only buffer one packet. Moreover, the nano-nodes have the ability to harvest energy form environment. In the network architecture, all the nano-nodes follow a random distribution. In the network layer, the different routing algorithms which have been investigated are implemented. In the MAC layer, a simple ALOHA type transmission scheme with positive and negative acknowledgement is installed. However, in flooding routing algorithm, nano-nodes send the packet directly without waiting for the ACK. In the physical layer, the Time Spread On-Off Keying (TS-OOK) modulation scheme is utilized, which is a common scheme in electromagnetic nanonetworks [22].

All parameters used in the simulations are listed in Table I. The packet send request interval ranges from 1-9 s, but nano-nodes only send packets in the case of sufficient energy and buffer. Due to the high density of nanonetworks, the packet Time To Live (TTL) is set to 50 hops. The pulse duration and pulse interarrival time is configured according to the parameters suggested in [10]. Due to the high path loss in THz band, the transmission range of nano-node should range from a few millimeters, hence, we simulate with different transmission ranges from 0.015-0.035 meters. The energy of sending and receiving packets and ACK/NACK depends on the size of the packets. Without loss of generality, we use unit energy to evaluate the energy consumption for sending and receiving packets. The values of learning, recovery and decay rate are referred to [23]. The simulations are conducted with different transmission ranges, densities of nano-node and energy harvesting speeds. The simulations for every set of parameters have been repeated for 5 times.

Moreover, we compare the proposed MDRQEN algorithm with random routing, flooding routing and the MDRQEN algorithm without the Q-learning update scheme. In the random routing algorithm, nano-nodes transmit packets to the next hop selected randomly from its neighbors. In the flooding routing algorithm, nano-nodes transmit packets to all the nano-node within its transmission range. In the MDRQEN algorithm

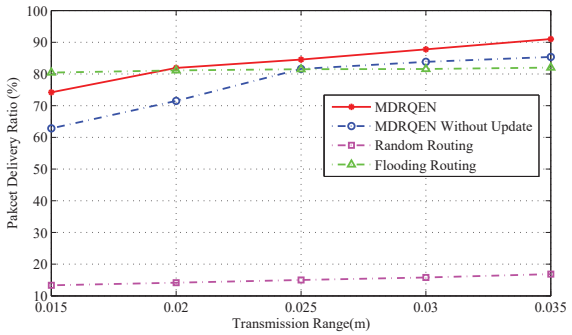

Fig. 2. Packet Delivery Ratio with Different Transmission Ranges

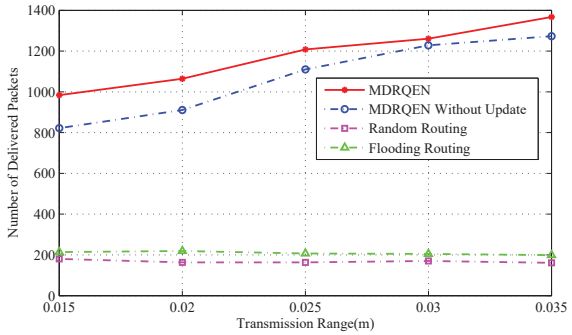

Fig. 3. Number of Delivered Packets with Different Transmission Ranges.

without the Q-learning update, the nano-nodes only add route entries into the routing table and choose the next hop nano-node randomly when the route entry is invalid.

C. Simulations with Different Transmission Ranges

The simulations with different transmission ranges are presented in Fig. 2, 3 and 4. Here, the density of nano-nodes is 12000 nodes/m^2, and the energy harvesting speed is 20 units/s.

It can be observed form Fig. 2 that: (i) the packet delivery ratio of all the routing algorithms increases with transmission range, because there are more chances to find a routing path to destination; (ii) the MDRQEN algorithm performs much better than the random routing algorithm. The reason is that the random routing algorithm chooses the next hop nano-node randomly from its neighbors, which leads to the probability of finding the right routing paths to the destinations be much lower than the MDRQEN algorithm; (iii) the packet delivery ratio of the MDRQEN algorithm is almost 10% higher than the MDRQEN algorithm without the Q-learning update scheme on average. Because the reward contained in the forwarded packet considers the packet deflected ratio, packet drop ratio, packet hop count and node energy status comprehensively, which results in a better routing selections or deflection decisions; (iv) when the transmission range is short, the packet delivery ratio of flooding routing algorithm is larger than the MDRQEN algorithm. Because the decrease of transmission range could enlarge the packet average hop count which can be observed from Fig. 3, and thus, increase the energy consumption of nano-nodes, which increases the packet loss probability due to the expiration of TTL and energy problem. Nevertheless, when transmission range increases, the packet delivery ratio of the MDRQEN algorithm becomes larger, since the MDRQEN algorithm can learn a more completely routing table and deflection table with long transmission range, while balancing the energy consumption and traffic load. Although the delivery ratio of the flooding routing algorithm is larger than the MDRQEN algorithm when transmission range is short, the number of delivered packets of the flooding routing algorithm is much smaller than the MDRQEN algorithm, which can be observed from Fig. 3. The reason is that, in the flooding

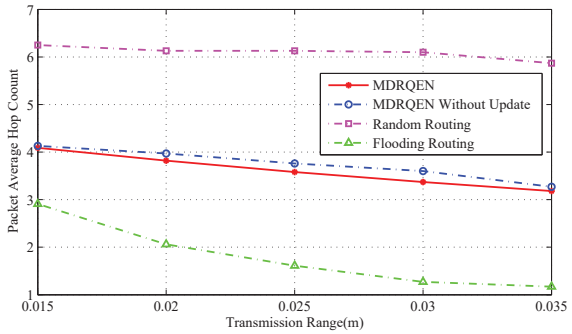

Fig. 4. Packet Average Hop count with Different Transmission ranges.

algorithm, nano-nodes spend most of its energy to flood the packets to find destinations, but without generating new packets, which indicates a lower network throughput and energy efficiency.

It also can be observed from Fig. 3 that the number of delivered packets of the MDRQEN algorithm (whether with Q-learning update scheme or not) increases with the transmission range. Because the routing table and deflection table can be updated more completely with large transmission range. On the contrary, for the random routing and flooding routing algorithm, bigger transmission range makes more nano-nodes join in finding the routing path to the destination rather than generating new packets. Hence, the number of delivered packets decreases with transmission range.

In Fig. 4, the packet average hop counts of all the routing algorithms are investigated with different transmission ranges. The increase of transmission range makes the nano-nodes more likely to find shorter routing paths for the packets, and thus, the packet average hop counts of all the routing algorithm decrease. The flooding routing algorithm has the lowest packet average hop count, since nano-nodes transmit the packets to all their neighbors within their transmission range, which make them easier to find shorter routing paths than other algorithms.

D. Simulations with Different Densities of Nano-nodes

The simulations with different densities of nano-node are presented in Fig. 5, 6 and 7. Here, the transmission range is 0.02 m, and the energy harvesting speed is 20 units/s.

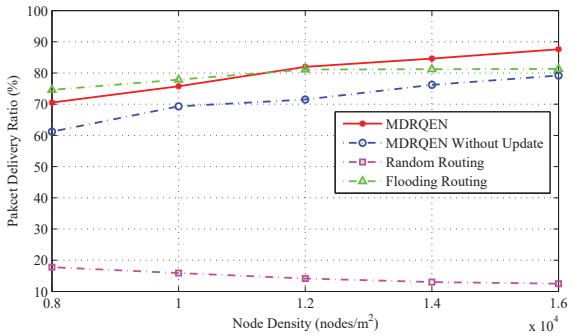

Fig. 5. Packet Delivery Ratio with Different Densities of Nano-nodes

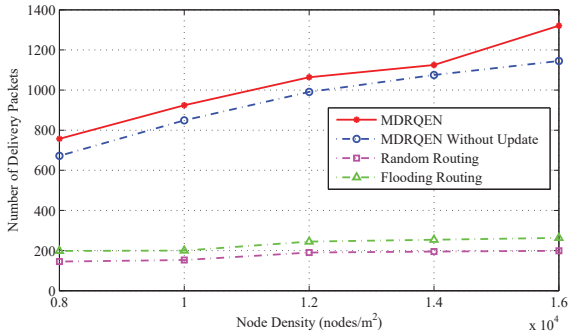

Fig. 6. Number of Delivered Packets with Different Densities of Nano-nodes

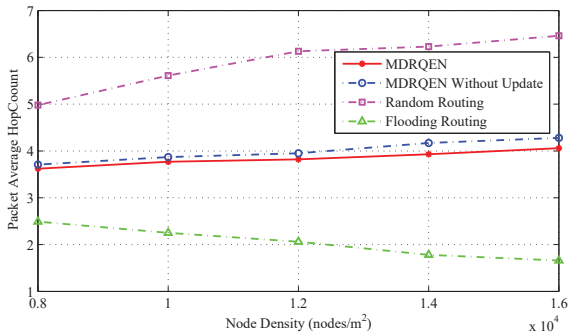

Fig. 7. Packet Average Hop Count with Different Densities of Nano-nodes

As shown in Fig. 5, the packet delivery ratio of the random routing algorithm decreases with nano-node density, because the probability of finding the right routing paths to destinations of packets decreases with the increase of nano-node quantity. However, the packet delivery ratios of the other three routing algorithms increase with the nano-node density. For the flooding routing algorithm, the reason is that more nano-nodes join in flooding the packets, which increases the probability of finding the right routing paths the destinations. For the MDRQEN algorithm, more nano-nodes help to update the routing table and deflection table to obtain the optimal routing paths to the destinations. Furthermore, when nano-node density is small, the packet delivery ratio of the flooding routing algorithm is lager than the MDRQEN algorithm. On one hand, the decrease of node density decreases the update frequency of the routing table and deflection table, which degrade the convergence of the MDRQEN algorithm. On the other hand, the decrease of nano-node density increases the relative distances between nano-nodes, and thus, increases the packet loss probability. However, the flooding routing algorithm allows nano-node reduce the packet loss probability by flooding the packets to all the nano-nodes within their transmission range.

Fig. 6 is plotted to investigate the number of delivered packets with different densities of nano-nodes. It can be observed that: (i) the number of delivered packets of all the routing algorithms increases with the density of nano-node. It is because more nano-nodes can generate more packets; (ii) the increase rate of delivered packets of the random routing and flooding routing algorithms are much lower than the MDRQEN routing. For the random routing algorithm, it is because the probability of finding the right routing path to destination decreases due to more nano-nodes. For the flooding routing algorithm, more nano-nodes join in forwarding the old packets without generating new packets. However, the MDRQEN algorithm can update the routing table and deflection table by the forwarded packets, hence, it can acheive a better performance than others.

Observed from Fig. 7, the packet average hop count of the flooding routing algorithm decreases with the node density. Because the increase of node density shorten the distances between nano-nodes, which make it easier to find the shorter routing paths to the destinations for packets. However, the other three algorithms only choose one next hop nano-node to forward the packet, hence, more nano-nodes indicates a smaller probability to find the shorter routing paths to destinations for packets.

E. Simulations with Different Energy Harvesting Speeds

The simulations with different energy harvesting speeds are presented in Fig. 8, 9 and 10. Here, the density of nano-node is 12000 nodes/m^2, and the transmission range is 0.02 m.

In Fig. 8, the packet delivery ratio is investigated with different energy harvesting speeds. The packet delivery ratios

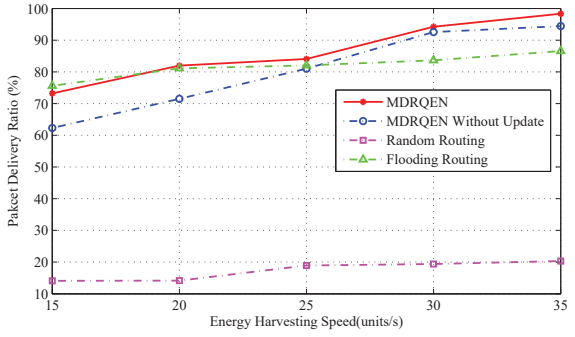

Fig. 8. Packet Delivery Ratio with Different Energy Harvesting Speeds

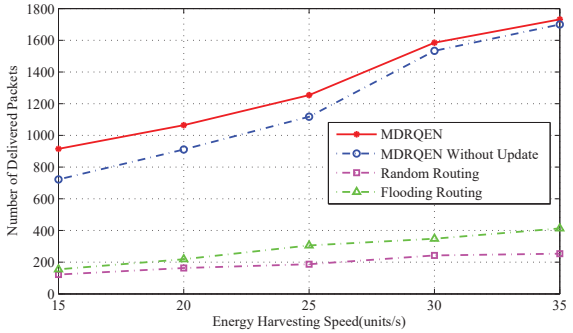

Fig. 9. Number of Delivered Packets with Different Energy Harvesting Speeds

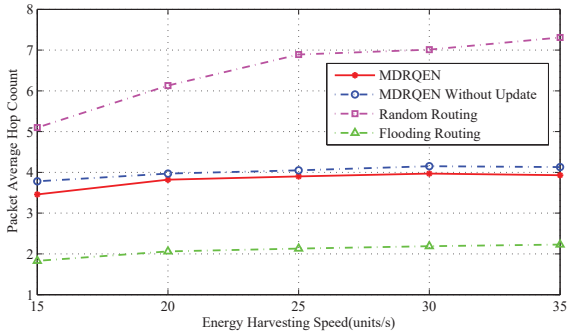

Fig. 10. Packet Average Hop Count with Different Energy Harvesting Speeds

of all the routing algorithm increase with the energy harvesting speed. The reason is that nano-nodes are able to transmit more hops to find the destinations with more energy. Furthermore, comparing to when energy harvesting speed is high, the MDRQEN algorithm is much better than the MDRQEN algorithm without the Q-learning update scheme when energy harvesting speed is low. Since in the proposed Q-learning update scheme, the energy statuses of nano-nodes are taken into consideration, which allows nano-nodes adapt to the change of energy statuses of the next hop nano-nodes, and thus, makes better foward/deflect decisions. However, nano-nodes still have the probability to forward/deflect the packets to the energy-less nano-nodes, which results in packets to be lost. While, flooding routing algorithm allows nano-nodes forward the packet to all the other nano-nodes within their transmission range to reduce such probability. When energy harvesting speed is low, the packet delivery ratio of the flooding routing algorithm is better than the MDRQEN algorithm. Nevertheless, when the energy harvesting speed enlarges, the MDRQEN algorithm can update the routing table and deflection table better, and thus, achieve a better performance.

As shown in Fig. 9 and 10, the number of delivered packet and packet average hop count of all the routing algorithms increase with energy harvesting speed. The reasons for the previous phenomenon is that nano-nodes are able to generate more packets with faster energy harvesting speed. The reason for the latter phenomenon is that nano-nodes can transmit more hops to find the destinations with faster energy harvesting speed.

IV. Conclusion

In this paper, a multi-hop deflection routing algorithm based on Q-learning for energy-harvesting nanonetworks is introduced to find the optimal routing paths to forward/deflect packets. In the algorithm, a deflction table is introduced to deflect the packet when the routing table entry is invalid due to energy or memeory/buffer issue. Moreover, a Q-learning update scheme is proposed to update the routing table and deflection table to adapt to the change of the network traffic loads and energy statuses of nano-nodes. In the Q-learning update scheme, the packet drop ratio, packet deflect ratio, packet hop count and node energy status are taken into consideration.

Comprehensively, the simulation results indicate that the proposed MDRQEN algorithm achieves the best performance by comparing to random routing algorithm, flooding routing algorithm and the MDRQEN algorithm without the Q-learning update scheme in terms of packet delivery ratio, number of delivered packet and packet average hop count. Although the packet average hop count of flooding routing algorithm is smaller than the MDRQEN algorithm, the number of delivered packets of flooding routing algorithm is much less than the MDRQEN algorithm, which indicates a lower network throughput and energy efficient of flooding routing algorithm.

In our future work, the effects of different learning parameters of the Q-learning update scheme will be studied, as well as the weights of different parameters in the reward.

Acknowledgment

This work is completed during Chaochao Wang's visit to Department of Electrical Engineering, University at Buffalo, The State University of New York. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61402414 and 61572438, and is part by Graduate Students International Training Fund of Zhejiang University of Technology.

References

[1] I. F. Akyildiz, F. Brunetti, and C. Blázquez, "Nanonetworks: A new communication paradigm," *Computer Networks*, vol. 52, no. 12, pp. 2260–2279, 2008.

[2] V. Rupani, S. Kargathara, and J. Sureja, "A review on wireless nanosensor networks based on electromagnetic communication." *International Journal of Computer Science and Information Technologies*, vol. 6, no. 2, pp. 1019–1022, 2015.

[3] J. M. Jornet and I. F. Akyildiz, "Graphene-based plasmonic nano-antenna for terahertz band communication in nanonetworks," *Selected Areas in Communications IEEE Journal on*, vol. 31, no. 12, pp. 685–694, 2013.

[4] Z. L. Wang, "Towards self-powered nanosystems: from nanogenerators to nanopiezotronics," *Advanced Functional Materials*, vol. 18, no. 22, pp. 3553–3567, 2008.

[5] S. Anand, D. S. Kumar, R. J. Wu, and M. Chavali, "Graphene nanoribbon based terahertz antenna on polyimide substrate," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 19, pp. 5546–5549, 2014.

[6] C. Rutherglen and P. Burke, "Nanoelectromagnetics: Circuit and electromagnetic properties of carbon nanotubes," *small*, vol. 5, no. 8, pp. 884–906, 2009.

[7] J. Kokkoniemi, J. Lehtomaki, K. Umebayashi, and M. Juntti, "Frequency and time domain channel models for nanonetworks in Terahertz band," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 2, pp. 678–691, 2015.

[8] M. A. Zainuddin, E. Dedu, and J. Bourgeois, "Low-weight code comparison for electromagnetic wireless nanocommunication," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 38–48, 2016.

[9] C.-C. Wang, X.-W. Yao, C. Han, and W.-L. Wang, "Interference and coverage analysis for terahertz band communication in nanonetworks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[10] J. M. Jornet, J. C. Pujol, and J. S. Pareta, "PHLAME: A physical layer aware MAC protocol for electromagnetic nanonetworks in the Terahertz band," *Elseiver Nano Communication Networks*, vol. 3, no. 1, pp. 74–81, 2012.

[11] S. Mohrehkesh, M. C. Weigle, and S. K. Das, "Drih-mac: A distributed receiver-initiated harvesting-aware mac for nanonetworks," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 1, pp. 97–110, 2015.

[12] S. DOro, L. Galluccio, G. Morabito, and S. Palazzo, "A timing channel-based mac protocol for energy-efficient nanonetworks," *Nano Communication Networks*, vol. 6, no. 2, pp. 39–50, 2015.

[13] S. Xu, B. J. Hansen, and Z. L. Wang, "Piezoelectric-nanowire-enabled power source for driving wireless microelectronics," *Nature communications*, vol. 1, p. 93, 2010.

[14] X. W. Yao, W. L. Wang, and S. H. Yang, "Joint parameter optimization for perpetual nanonetworks and maximum network capacity," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 4, pp. 321–330, 2015.

[15] A. Tsioliaridou, C. Liaskos, S. Ioannidis, and A. Pitsillides, "Corona: A coordinate and routing system for nanonetworks," in *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*. ACM, 2015, p. 18.

[16] C. Liaskos, A. Tsioliaridou, S. Ioannidis, N. Kantartzis, and A. Pitsillides, "A deployable routing system for nanonetworks," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[17] J. Xu, R. Zhang, and Z. Wang, "An energy efficient multi-hop routing protocol for terahertz wireless nanosensor networks," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2016, pp. 367–376.

[18] A. Zalesky, H. Le Vu, M. Zukerman, Z. Rosberg, and E. W. Wong, "Evaluation of limited wavelength conversion and deflection routing as methods to reduce blocking probability in optical burst switched networks," in *Communications, 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 1543–1547.

[19] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[20] A. Belbekkouche, A. Hafid, and M. Gendreau, "Novel reinforcement learning-based approaches to reduce loss probability in buffer-less obs networks," *Computer Networks*, vol. 53, no. 12, pp. 2091–2105, 2009.

[21] S. Haeri, W. W.-K. Thong, G. Chen, and L. Trajković, "A reinforcement learning-based algorithm for deflection routing in optical burst-switched networks," in *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*. IEEE, 2013, pp. 474–481.

[22] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1742–1754, 2014.

[23] S. P. Choi and D.-Y. Yeung, "Predictive q-routing: A memory-based reinforcement learning approach to adaptive traffic control," in *Advances in Neural Information Processing Systems*, 1996, pp. 945–951.