Energy-Efficient Routing and Rate Allocation for Delay Tolerant Networks

Yuanyuan Zeng
School of Electronic Information
Wuhan University
Wuhan, China
zengyy@whu.edu.cn

Jie Wu
Department of Computer and Information Sciences
Temple University
USA
jiewu@temple.edu

Naixue Xiong
Department of Computer Science
Georgia State University
USA
nxiong@cs.gsu.edu

Deshi Li
School of Electronic Information
Wuhan University
Wuhan, China
dsli@whu.edu.cn

***Abstract*—Routing for Delay Tolerant Networks (DTNs) are challengeable for the continuously varied network environment. Most of existing DTN routing algorithms mainly focus on metrics such as delay, hop count and bandwidth, etc. Green communication is a new focus with the goal of saving energy by optimizing network performance and ultimately protecting the natural climate. In this paper, we present an Energy-efficient Routing and Rate Allocation (ERRA) scheme based on Q-learning that can optimize the energy efficiency with the constraints of congestion, buffer and delay. ERRA solves the routing and rate allocation together with reinforce leaning, and then make decisions on relay selection and rate schedule. ERRA explores the possible strategies, and then exploits the knowledge obtained to adapt its relay and schedule strategies. ERRA achieves the desired overall objective by considering the stochastic non-cooperative game under on-line multi-commodity routing situations. The simulation results show that ERRA achieves good energy efficiency and delivery ratio within the delay bound.**

Keywords- DTN; Routing; Rate allocation; Energy-efficient

I. INTRODUCTION

Delay Tolerant networks enable opportunistic intermittent communication when mobile nodes are connected only intermittently. The DTN topology is dynamic, since the network lacks continuous connectivity and may be partitioned at any instant. The uncertainty of network environment in DTNs is a result of the mobility, limit wireless radio range, sparsity of mobile nodes, energy resources, etc. [1]. Considering intermittent communications from end-to-end paths, routing in DTN takes a "carry and forward" approach [2] that store and carry the data locally, then eventually delivering it either to the destination or to a relay deemed to meet the destination sooner.

The existing DTN routing protocols mainly focus on the schemes to increase the likelihood of finding opportunistic paths. But, it is difficult to model the DTNs and their possible connections in realistic network situations. The methods of replicate data packets can improve the possibility of opportunistic connections, but it also brings burden for storage and bandwidth resources. In order to perform efficient routing in DTN, nodes need to predict the varied network environment and wisely allocate network resources. Existing routing schemes mainly utilize metrics such as delay, hop count and bandwidth, in traffic data delivery. In previous related work, energy efficiency is not the main concern in DTNs. Nowadays, green ICT (Information and communication technology) is focused for reported high carbon emissions each year. We present an energy-efficient routing and rate allocation (ERRA) algorithm, which is designed to explicitly optimize the energy efficiency with constraints of congestion, buffer and delay bound. The main contributions of this paper include:

- ERRA can carry and intelligently "pull" traffic data toward the optimal direction with rate allocation by predicting the network environment with reinforce learning of the distributed network systems.
- ERRA achieves energy-efficient routing paths with the constraints on congestion, buffer and delay bound.
- ERRA uses multi-agent Q-learning approach for on-line multi-commodity routing situations and provides the optimal routing and rate scheduling for each communication pair by Nash equilibrium strategy.

Section II presents the related work. Section III presents some definitions and models. We present details of ERRA in Section IV. Section V is simulation results. Section VI concludes this paper.

II. BACKGROUND AND RELATED WORK

The existing routing schemes in DTNs can be classified into two categories that are based on replication and forwarding, respectively.

Replication schemes are to replicate copies of a packet in the hope that it will succeed in reaching the destination. The schemes are commonly used to maximize the probability of a packet being successfully transferred. The kinds of routing schemes differ in the replication models and ways to cut down the replication overhead. Epidemic routing protocol [3] uses a naive flooding method, but it wastes resources and degrades the network performance. Spray and Wait routing scheme [4] replicates the packet by bounding the number of

1545-0678/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDCSW.2012.69

replicas, but it is feasible with large amounts of local storage and enough bandwidth. There are some routing schemes that limit the packet replication with the consideration of storage constraints [5,6]. RAPID [7] is an intentional DTN routing protocol, which treats routing as a resource allocation problem and decides on packet replication according to per-packet utilities. Replication schemes increase the opportunistic communication probability with the tradeoff of big burden on network resources like available storage and link bandwidth.

Forwarding schemes maintain at most one copy of a packet in the network during routing process. The forwarding based routing schemes forward the packet toward the direction of optimal routing metrics. Jain et al [8] proposed a modified Dijkstra algorithm based on a time-dependent Graph. Hay et al [9] proposed a deterministic DTN routing and scheduling when the contact times between nodes are known in advance or can be predicted. Dvir et al [10] proposed a dynamic backpressure routing in DTNs. Recently, intelligent algorithms are utilized in DTN routing. Ahmed et al [11] proposed a Bayesian classifier based DTN routing framework to infer the optimized routing. Huang et al [12] proposed a fuzzy logic based routing in DTNs, which is exploited to select the close-by intermediate node on the path to the destination.

Recently, green communication and computing have been proposed as a solution to addressing the growing cost and environmental impact of telecommunications, in which energy efficient schemes provide positive solutions. Sanctis et al [13] discuss several techniques on energy efficient wireless networks towards green communications and outline challenges and open issues.

III. PRELIMINARIES

We assume that: each node gets knowledge of its locations and has specific traveling trajectory. There are some static access points are deployed in DTN network that help to make connections and disseminate some network routing information.

A. Energy Consumption for Communications

As in usual network communication, each node can be in one of the three working states: listening, send and receive data. ERRA is helped to choose route and rate for transmissions at the beginning of each frame. If no events occur on current node, it can tune into sleep state to save energy.

The energy consumption of DTN nodes include energy consumed for transmitting E_t, receiving E_r and listening E_l. Here, we omit the energy consumption of sleep state. The energy consumption can be calculated as Formula (1), where e_e is the energy consumed by transceivers per second, and e_a is the energy consumed in the transmitter RF amplifier per second; e_p is the energy consumed for processing in the receivers, and e_l is the energy consumed for listing to the radio environment. e_l equals e_e. Those parameters are determined by the design characteristics of transceivers. R is the transmission range; n is the power index of the channel path loss. T_t is the time for sending data. T_r is the time for receiving. T_l is the listening time. T is the time length of a cycle.

$$\begin{aligned} E_t &= (e_e + e_a R^n) T_t \\ E_r &= (e_e + e_p) T_r \\ E_l &= e_l T_l = e_l (T - T_t - T_r) \end{aligned} \tag{1}$$

B. Rate and buffer

We assume communication time is divided into continuous equal frames, which can be divided into continuous equal slots. Each node has the same frame structure and chooses transmit rate according to channel congestion and energy efficiency. The maximum rate of node u can be determined from Formula (2). In the formula, C is the channel capacity. W is the bandwidth and P is transmission power; g is the channel gain, and N_0 is the noise power spectrum density; and F is the gap to ergodic channel capacity.

$$rate_t(u) = \frac{T_c}{T} C = \frac{T_c}{T} W \log_2 \left(1 + \frac{Pg}{N_0 WF}\right) \tag{2}$$

In which, T is the time of a frame. T_c is the opportunistic contacting time of two nodes in a frame, where T_c is related with contacting topology changes and rate constraints. It is obvious that T_c/T is proportional to the schedule slot numbers.

The sending and receiving rate on node u can be formulated as:

$$rate(u) = rate_t(u) + rate_r(u) = \frac{T_t + T_r}{T} C \stackrel{\Delta}{=} (a_t + a_r) * C \tag{3}$$

In which, $(T_t + T_r)/T$ is the active fraction of slot allocation on current node for sending and receiving data. There is: $T_t + T_r = T_c$. We use λ_t and λ_r to represent the slot allocated fraction for sending and receiving data, respectively.

Considering a half-duplex network interface card for node-to-node communications, there is:

$$\lambda_t(u) + \lambda_r(u) \le 1 \tag{4}$$

Considering the congestions, for slot τ, it should satisfy Formula (5), where u' is in the interference node set $I(u)$ of node u:

$$\lambda_t(u) + \sum_{u' \in I(u)} \lambda_t(u') \le 1 \tag{5}$$

According to interference model, a node receiving data from a neighbor should be spatially separated from any other transmitter by at least a distance D, i.e., interference range. If distance between u and u' is less than D, then the two nodes interfere with the transmission.

Each node has a buffer to store and carry the packets that have not been transmitted. The buffer has limited maximal size and can contain packets with its available space, e.g., node u has available buffer size as $buf(u)$. Let $occ(u, \tau)$ be the number of bytes stored at node u at slot τ under current routing strategies.

For all u, τ, there is:

$$occ(u,\tau)\le buf(u) \tag{6}$$

C. Delay Bound

A delay bound T_max is given to limit the maximal delivery time for traffic packets. For each node u of the route path with the h^{th} hop in the network, the distance of itself from the destination dst can be measured by Euclid distance as: $d=\|u-dst\|$. Note that dst can be an average node or an access point.

So an approximate average delivery velocity $\bar{v}$ needed is calculated as:

$$\bar{v}=\frac{d}{slack}=\frac{\|u-dst\|}{T\_\max-\sum_{i=1}^{h}T_i} \tag{7}$$

In which, *slack* is the time left for routing, which is equal to the remaining part of delay bound minus cumulative time on each hop for the packet.

D. Problem formulation

The problem to find feasible routing and rate allocation with maximal delay bound can be formulated as:

- **Objective**: The objective is to minimize the total routing and rate allocation cost on the route path, i.e., to minimize the cost of cumulative hops h on the route path as shown in Formula (9).

$$\check{R}(S)=\sum_{1}^{h}\check{R}(u) \tag{9}$$

- **Rate constraints**: The corresponding rate on the path route should satisfy the interference constraint in Formula (4) and (5).
- **Buffer constraints**: The relay node on the path should satisfy the buffer requirement in Formula (6).
- **Delay constraints:** The relay node on the each hop should satisfy the delivery velocity in Formula (7).

IV. ERRA Scheme

A. Q-learning approach

The objective of ERRA is to learn from the environment states (dynamic connection events) and decide on actions, so as to maximize the reward i.e., minimize the cost function. The DTN control system is formulated by a tuple $<S, A, \check{R}, \check{T}>$, where S is the discrete hazard state space. A is the discrete action space that is dependent on strategies taken. $\check{R}$: $S\times A\rightarrow \mathbf{R}$ is the cost function, which implies the quality of a state-action combination of the network system. $\check{T}$: $S\times A \rightarrow \Delta S$ is the state transition function, where ΔS is the probability distribution over state space S.

In ERRA, the actions include sets of selected route and rate strategies, i.e., a set of tuple pairs as: (*relay*(u), *rate*(u)). Once an action (i.e., with a specific strategy) is taken, the network system produces new performance signal (i.e., cost) according to it. Then ERRA receives the update cost $\check{R}$, which is used to evaluate the effectiveness of the action. The learning procedure is achieved by updating the Q-value. The Q-learning approach converges to an optimal strategy as long as the state-action pairs are continually updated. In scenarios discussed in the paper, each node will learn and predict the optimal routing relay and schedule through reinforce learning process. When traffic relay transmission beginning at time slot τ is finished at the next time $\tau+\lambda$ (among it, λ is the time slots for transmission schedule), then the Q-value for state-strategy pair is updated by Formula (10).

$$Q'(s,a)=(1-\alpha)Q(s,a)+\alpha(\check{R}_\tau+\beta\min_{\alpha}Q(s',a')) \tag{10}$$

In Formula (10), α is the learning rate and in the range of (0,1). β is the discount factor and is in the range of (0,1) too. We use a constant learning factor, and the learning procedure can track the dynamic network situations.

B. Cost function

The ERRA scheme aims to provide energy-efficient routing and rate allocation. The cost function is achieved by the average amount of energy consumption per bit on the current route path from 1 to h hops, which embodies the tradeoff among energy efficiency, connection duration and communication efficiency:

$$\check{R}(path)=\sum_{1}^{h}\check{R}(u)=\sum_{1}^{h}\frac{E}{rate(u)*T_c}=\sum_{1}^{h}\frac{E_t+E_r+E_l}{rate(u)*T_c}$$
$$=\sum_{1}^{h}\frac{(e_e+e_aR^n)\lambda_t+(e_e+e_p)\lambda_r+e_l(1-\lambda_t-\lambda_r)}{(\lambda_t+\lambda_r)C*(\lambda_t+\lambda_r)} \tag{11}$$

Considering the rate constraints and buffer constraints, we formally express the cost function of ERRA strategy S as Formula (12).

$$\check{R}(u)=\begin{cases}\dfrac{(e_e+e_aR^n)\lambda_t+(e_e+e_p)\lambda_r+e_l(1-\lambda_t-\lambda_r)}{(\lambda_t+\lambda_r)C*(\lambda_t+\lambda_r)} & if\ satisfies (4)-(7)\\ \infty & otherwise\end{cases} \tag{12}$$

For one-commodity routing situation, if node $u\notin\{s, d\}$ on the route, then the traffic data bits sent out equal to the received data bits, i.e., $\lambda_t=\lambda_r$. The received traffic rate depends on the traffic rate sent out of the last hop on the route.

C. Nash Q-learning approach

For on-line multi-commodity cases, Q-learning approach is extended for multi-agent decision making. In on-line multi-commodity situations, each agent is selfish for routing desire. The network routing and allocation process is modeled as a stochastic non-cooperative game framework. Let $<N, \{S\}, \{\check{R}\}>$ denote the stochastic non-cooperative routing and rate allocation game, where N is the number of routing commodities in the network, $\{S\}$ is the strategy set, and $\{\check{R}\}$ is the cost set. The objective of multi-commodity routing is to minimize the overall cost. Different agents have their own objectives which could be conflicting with each other, so the overall cost is dependent on the strategy selection of each agent. The strategy of each agent is responsible for selecting routes and rates (i.e., transmitting slots) that the overall multi-commodity routing strategy cost achieves minimized cost during the learning process. Considering the multi-commodity situation, multi-agent Q-learning approach is utilized to find out the optimized ERRA strategy to achieve the minimized cost for all route paths. We consider $\pi_i(S)$ as the probability for the i^{th} routing commodity that selects a specific strategy S at time τ with the game equilibrium. Π_i is the set of possible strategies. The overall objective in the stochastic non-cooperative routing and rate allocation game can be express as:

$$\min_{\pi_i \in \Pi_i} \quad E[\check{R}_i(S)] \ \ for \ \ all \ \ i \tag{13}$$

Where:

$$E[\check{R}_i(S)] = \sum_i \pi_i(S)\,\check{R}_i(S) \tag{14}$$

In the stochastic non-cooperative game based network system, each routing commodity finds a strategy with Nash equilibrium to achieve objective in Formula (13).

Let $NashQ_i$ be agent i's cost in current state with the selected equilibrium. According to [14], the Nash equilibrium $NashQ=(\pi_1^*,\ldots\pi_n^*)$ is computed from Formula (15). In the formula, m is the number for players in the game. In order to calculate the Nash equilibrium, each agent i need to know the other agents' Q-values. Then, each agent observes the other agents' immediate costs and actions. So, agent i can update its Q-value according to other agents' Q-values as shown in Formula (16). In each time step at time τ, a player observes the current state s, and then takes action a. An immediate cost $\check{R}$ and the next state s' are observed.

$$NashQ_i(s,a) = \sum_{1}^{m} \pi_i^*(s,a) Q_i(s,a) \tag{15}$$

$$\begin{aligned} Q_i'(s,a_1,\ldots,a_m) &= (1-\alpha)Q_i(s\ ,a_1,\ldots,a_m) \\ &+ \alpha(\check{R}_i + \beta NashQ_i(s'\ ,a_1,\ldots,a_m)) \end{aligned} \tag{16}$$

To minimize the network cost, the multi-agent Q-learning approach has to explore all possible strategies randomly and greedily, and then chooses the "good" strategy. The strategy exploration probability is updated as shown in Formula (17). γ is a constant factor between 0 and 1. The learning policy satisfies the GLIE (Greedy in the Limit with Infinite Exploration) property.

$$\pi_i'(a^*) = \begin{cases} \pi_i(a) + \gamma(1-\pi_i(a)), & if\ a^* = \arg_a NashQ_i(s,a) \\ \gamma\pi_i(a), & if\ a^* \neq \arg_a NashQ_i(s,a) \end{cases} \tag{17}$$

The Nash Q-learning based ERRA algorithm is described in Algorithm 1. Line 1-6 is the algorithm initialization. In line 4, $|A|$ is the number of possible strategies, which is bounded by the multiplication of maximal degree of the network graph Δ and frame slot number L, i.e., $|A|=\Delta*L$. Line 7-20 is the Nash Q-learning procedure. In line 13, $\check{R}_1\ldots\check{R}_m$ represent the cost for all players, and $a_1\ldots\ a_m$ represent the strategy taken by the other players except a_i. In Line 14, slot time τ is updated as $(\tau+\lambda)$ mod L, where L is the frame length. Line 16 shows the Q-value update of each user for its next state according to Formula (16). As explained in [15], the time complexity and space requirement of this learning algorithm is high when agent number is big. For 2-player Nash Q-learning, it has exponential worst-case time complexity. The space complexity is also exponential in the number of users. In the network, the game of routing resources occurs among the routing commodities within the interference range during the contacting time. The routes diverged from interference range will not affect one another. So the routing game process of the network system can be achieved by the local game with local routing commodities, when there are joints routes within the interference range from one another during the contacting time.

Algorithm 1: Nash Q-learning based Algorithm

1 **for** i=1…m //m agents with src_i and dst_i
2 Let τ=0, get the initial state s=$s(\tau)$
3 **for** all s S and a A
4 $Q_i(s, a^1,\ldots, a^m) = 0, \pi_0^i(a) = 1/|A|$
5 **endfor**
6 **endfor**
7 **while** (network execution condition is TRUE)
8 $u_i = src_i$
9 **for** current node u_i //u_i is the current hop for i^{th} agent
10 **if** ($u_i == dst_i$) break;
11 **endif**
12 Choose action a_i according to (17)
13 Take a_i for $s(\tau)$ and observe $\check{R}_1\ldots\ \check{R}_m$, $a_1\ldots\ a_m$
14 update τ =(τ+λ) mod L, the next state s'=$s(\tau)$
15 **for** all $j\neq i$, j=1…m
16 Update Q_j according to (16)
17 **endfor**
18 s=s', u_i=$a(relay(u_i))$//update state and the next hop
19 **endfor**
20 **endwhile**

The local game algorithm is illustrated in Algorithm 2. In Line 14, j is the commodity with routes that are within the interference range of i for current contacting state by satisfies: $\|route(i)\text{-}route(j)\|\leq D$. The distance of the routes are defined as the minimal distances of the relay nodes from the two routes respectively. So we only observe the cost and action of contending commodity with current commodity i, but not all the other commodity in the network. In a general DTN, the network is usually sparse or loosely connected. The on-line routing commodity number is often small within the interference range, so the performance of local game based ERRA is acceptable.

Algorithm 2: Local game based ERRA

for i=1…m //m agents with src_i and dst_i
  let τ=0, get the initial state s=$s(\tau)$
  for all s S and a A
    $Q_i(s, a^1, ..., a^m) = 0, \pi_0^i(a) = 1/|A|$
  endfor
endfor
while (network execution condition is TRUE)
  $u_i = src_i$
  for current node u_i // u_i is the current hop for i^{th} agent
    if ($u_i == dst_i$) break;
    endif
    choose action a_i according to (17)
    take action a_i for $s(\tau)$ and get $\check{R}_i$
    observe $\check{R}_j$, a_j ($j \neq i$, j=1…m), if $\|route(i)\text{-}route(j)\| \le D$
    update $\tau = \tau + \lambda$, the next state $s' = s(\tau)$
    for all j //j represent the local contender
      Update Q_j according to (16)
    endfor
    $s=s'$, $u_i = a(relay(u_i))$//update state and the next hop
  endfor
endwhile

V. SIMULATIONS

We simulate an ad hoc delay tolerant network with 30 nodes around 500m× 500m area. The nodes travel with speed from 0 to maximal 2.5m/s around the test area. A predefined moving model is given for each node. A fixed static node is placed at the center of the area, which is used to simulate an access point in real networks. In the simulations, the nodes connect with each other by a half-duplex wireless network interface card. The communication range is 15m, while the interference range is set to 30m. At the beginning of the simulation, the routing request with CBR traffic is periodically generated from a source node that is randomly selected. The traffic packet interval is 500ms. By default, each node allocates a limited buffer with maximal 5 packets in buffer. The simulated access point is fixed chosen as routing destination. We initialize the parameters: α=0.1, β=0.9 and γ=0.5. The value of energy consumption related parameter e_e, e_a, e_p, and e_l is chosen based on [15].

We make simulations when the routing commodity number is 1, 2, and 4 respectively. In 4-commodity situations, we use local game based algorithm.

A. Convergence

The result is shown in Figure 1. It shows the average path cost from 10s to 250s during the simulation. We observe that the average path cost begins to converge around 200s for all the situations. Before the convergence point, there are fluctuates existed for the learning process. When comparing multi-agent results with 1-agent situation, they seem to have more path cost. This is because Nash equilibrium strategy is chosen to provide optimal routing cost for all the agents in the network, which may not be the optimal strategy for each single agent.

B. Learning ability

We then make simulations with the varied network scenarios. The first scenario is: we add 5 more default nodes into the network to help the connections at 60s after the simulation begins, and then we remove them at 120s. The corresponding results are shown in Figure 2. We observe that ERRA can track the change and adapt the strategies toward the network environment. The average path cost of routing situations with 5 more nodes added into the network is a little lower than the original situations. It implies that the added nodes help to improve the contacting opportunity in the network without extra congestions because of effective rate allocation control. At 120s of the simulations, we remove those added nodes. It incurs a few fluctuations of average path cost, because each agent has to predict and learn new strategy toward the changes. The second scenario is: we half the routing traffic interval from 60s of the simulations, and then recover it at 120s. The corresponding results are shown in Figure 3. We observe that the average path cost increases a little from 60s to 120s when compared with the original results. This is because the increased traffic load incurs possible congestions. According to ERRA, the routing and rate strategies will adapt to improve the cost. Then more idle energy consumption needed in long schedule for the increased traffic. After 120s, the traffic load is recovered to the original situations. The results show there are many fluctuations after 120s, because ERRA learns and select new strategies toward the current the new network environment.

C. Delay-bounded delivery

Figure 4 shows the data average real-time delivery ratio as delay bound increases from 200ms to 2000ms, which is based on results of 250s simulations. We observe that the data delivery ratio increases as relax the delay bound. And 1-agent achieves better delivery when compared with 2-agent and 4-agent situations. This is because more routing agents bring more possible congestions in the networks and the need more schedule time for transmission according to ERRA rate allocation.

D. Comparisons

We then compare ERRA with Epidemic [3] and modified Dijkstra algorithm [8]. Firstly, we simulate under network scenario 1 with 2-commodity routing situations: add 5 more default nodes into the network at 60s after the simulation begins, and then remove them at 120s. Figure 5 shows the average energy consumption per bit during the simulations. From the results, we can see that ERRA has much better energy efficiency when compared with the other two algorithms, especially when network topology changes from

60s to 120s. Epidemic algorithm uses replication to deliver data packets in the network, so the more energy is needed. And this is even worse when simulation goes on, because much replicated forwarding occurs during the packet delivery. The modified Dijkstra algorithm uses forward method to minimize the average delivery delay, but it does not consider the energy consumption of packets and cannot adapt toward the varied network environment. When more nodes are added into the networks, the modified Dijkstra consumes more energy for increased congestions. Since we simulate the 2-commodity situation, the increased congestions are not big, so the a little more energy needed to consume for the varied topology. Figure 6 shows the energy consumption per bit with scenario 2: we half the routing traffic interval from 60s of the simulations, and then recover it at 120s. The results show similar tendency as in scenario 1. More energy consumptions in Epidemic and modified Dijkstra algorithm, because increased traffic load brings more congestions and waste energy for packets.

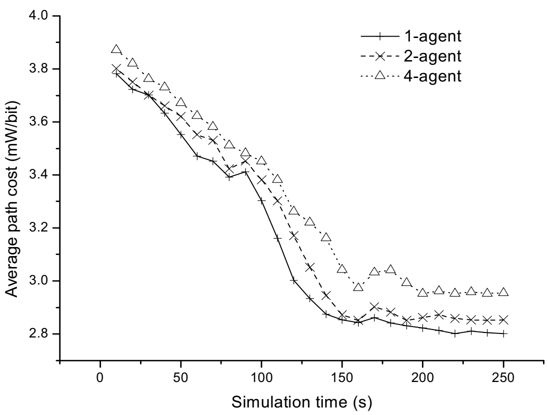

Figure 1. Convergence.

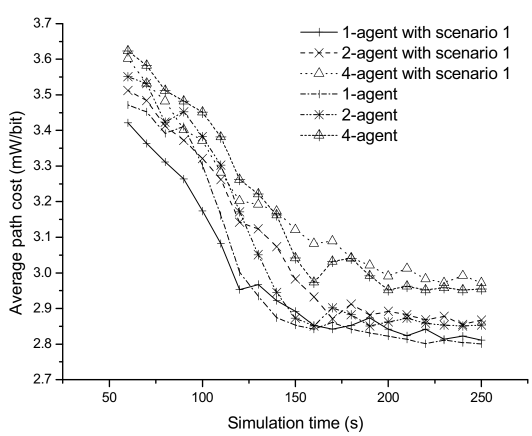

Figure 2. Learning ability with scenario 1.

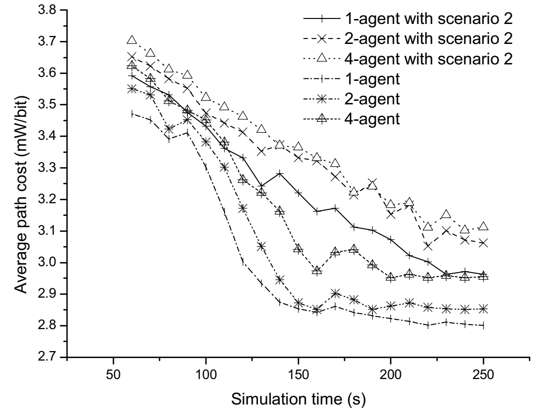

Figure 3. Learning ability with scenario 2.

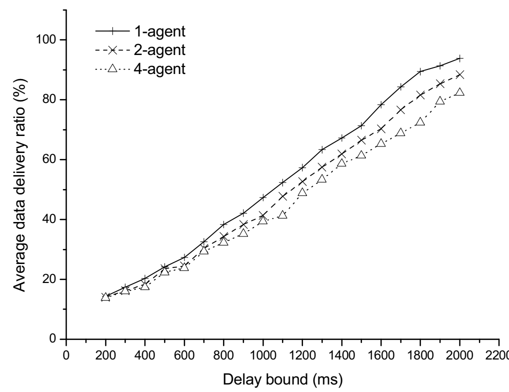

Figure 4. Data delivery ratio.

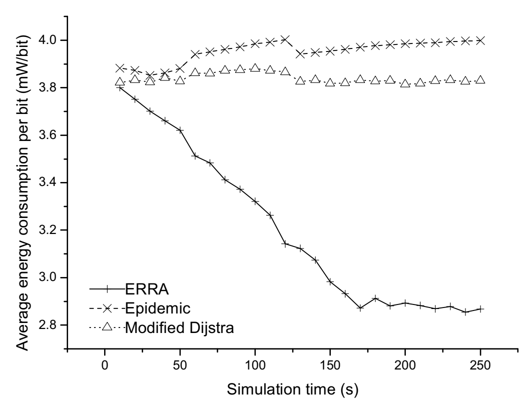

Figure 5. Ave. energy consumption in scenario 1.

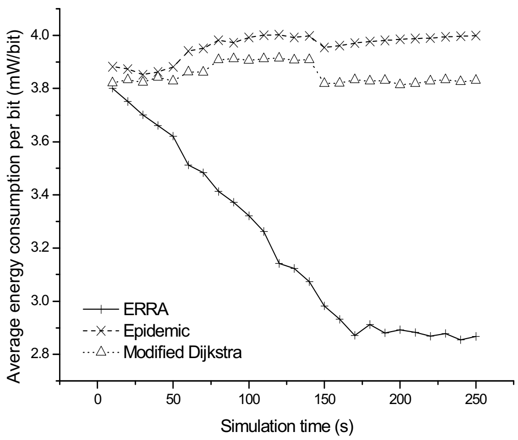

Figure 6. Ave. energy consumption in scenario 2.

VI. CONCLUSION

Considering necessity for green communication and computing, we make research on energy-efficient routing and rate allocation in delay tolerant networks. We propose to utilize Q-learning approach to "pull" the data packets toward optimal routing direction with optimal schedule by predicting the unknown network environment with opportunistic communications. Our ERRA provide energy efficient, less-congested and delay-bounded data delivery. We make simulations on DTN networks and make performance evaluations. The results show ERRA can effectively improve the energy efficiency and data delivery ratio within unknown and varied network situations.

ACKNOWLEDGMENT

This work was supported by Chinese National Science Foundation Project (no. 61103218), Hubei Province National Science Foundation Project (no.2011CDB446), Chinese National Program on Key Basic Research Project (no. 2011CB707106) and Wuhan University Automatic Research Project (no.111093).

REFERENCES

[1] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, 16(8), pp. 2349-2370, 2010.

[2] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks", *IEEE/ACM Transaction on Networking*, 10(4), pp. 477-486, 2002.

[3] W. Mitchener and A. Vahat, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report CS-2000-06, Duke University, 2000.

[4] T. Spyropoulous, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An efficient routing scheme for intermittently connected mobile netowrks," *ACM WDTN*, 2005.

[5] T. Small and Z. Haas, "Resource and Perforamance Tradeoffs in Delay-Tolerant Wireless Networks," *ACM SIGCOMM workshop on Delay-tolerant networking(WDTN)*, 2005.

[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption- Tolerant Networks", *IEEE Infocom*, 2006.

[7] A. Balasubramani, B. N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," *ACM SIGCOMM*, 2007.

[8] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network,"*ACM SIGCOMM*, 2004.

[9] D. Hay, P. Giaccone, "Optimal routing and scheduling for deterministic delay tolerant networks," *IEEE WONS*, 2009.

[10] A. Dvir, A. V. Vasilakos, "Backpressure-based routing protocol for DTNs," *ACM SIGCOMM*, 2010.

[11] S. Ahmed, S. S. Kanhere, "A Bayeisan Routing Framework for Delay Tolerant Netowrks," *Wireless Communications and Networking Conference (WCNC)*, 2010.

[12] C.-J. Huang, H.-Y. Shen, J.-J. Liao, K.-W. Hu, D.-X. Yang, C.-H. Chen, and Y. –T. Chuang, "A fuzzy logic-based routing for delay-tolerant heterogeneous networks," *IEEE International Conference on Granular Computing (GRC)*, 2009.

[13] M. D. Sanctis, E. Cianca, V. Joshi, "Energy Efficient Wireless Networks Towards Green Communications," *Wireless Personal Communictions*, pp: 1-16, Feb. 2011.

[14] J. Hu and M. P. Wellman, "Nash Q-learning for General-sum Stochastic Games," *Journal of Machine Learning Research*, 4, pp: 1039-1069, 2003.

[15] W. Feng and J. M. H. Elmirghani, "Green ICT: energy efficiency in a motorway model," *3rd International Conference on Next Generation Mobile Applications, Servies, and Technologies*, 2009.