Accelerated Ants Routing in Dynamic Networks

Hiroshi Matsuo, Kouichi Mori
Nagoya Institute of Technology, Gokiso, Nagoya, 466-8555, JAPAN
matsuo@elcom.nitech.ac.jp

Abstract

Efficiently routing in a dynamic network is a important problem in ad hoc network according to development of personal data assistant (PDA) and wireless network equipment. However conventional routing algorithm is difficult to apply to dynamic topology network. Q-Routing, DRQ-Routing and Ants-Routing which are based on reinforcement learning technique are proposed. But convergence speed and routing result are still unsatisfied. In this paper, accelerated Ants-Routing which increase convergence speed and obtain good routing path is discussed. Experiment on dynamic network showed that accelerated Ants-Routing learns the optimum routing in terms of convergence speed and average packet latency.

1. Introduction

Efficiently routing in a dynamic network is an important problem in ad hoc network according to the development of personal data assistant (PDA). Features of ad hoc network are frequent and unpredictable changes in network topology. There is no center server which knows all node information. Therefore, it is not appropriate to apply conventional Internet routing algorithms such as RIP or OSPF[1]. These algorithms are based on minimizing the number of hops which means number of relay node between source node and destination node. But changing topology generates a lot of routing information packet (called flooding in OSPF) and a lot of time is needed to converge routing informations.

In order to solve these problems, routing algorithm based on reinforcement learning are proposed. These algorithms are only used local information to decide routing table in each intermediate nodes. Boyan and Littman proposed Q-Routing which use the Q-Learning framework to solve these dynamic routing problem[2]. Kumar and Miikkulainen proposed Dual Reinforcement Q-Routing[3]. Subramanian proposed Ants-Routing based on simple biological "ants" that explore the network and rapidly learn good routes, using a novel variation of reinforcement learning[5].

This paper presents a modified Ants-Routing algorithm called accelerated Ants-Routing which makes convergence speed accelerated compared with other reinforcement base algorithms mentioned above.

This paper is organized as follows. In Section 2, we describe the reinforcement learning and Q-Learning algorithm. In section 3, we describe three typical routing algorithm based on reinforcement learning, Q-Routing, DRQ-Routing and Ants-Routing. In section 4, accelerated Ants-Routing algorithm is proposed. And in section 5, convergence speed of Q-Routing, DRQ-Routing, Ants-Routing and Accelerated Ants-Routing are compared in various dynamic topology changing environment.

2. Reinforcement Learning

2.1. Framework of Reinforcement Learning

Reinforcement Learning is the process by which an agent improves its behavior in an environment via experience. The feedback of an environment is simply a scalar value which may be delayed in time. This reinforcement signal reflects the success or failure of the entire system after it has performed some sequence of actions. Hence the reinforcement signal does not assign credit or blame to any one action, or to any particular node or system element.

In contrast, in supervised learning the feedback is available after each system action, removing the temporal credit assignment problem; in addition, it indicates the error of individual nodes instead of simply telling how good the outcome was. Supervised learning methods, for instance backpropagation, rely on having error signals for the system's output nodes, and typically train on a fixed set of examples which is known in advance. But not all learning problems fit this paradigm. Reinforcement learning methods are appropriate when the system is required to learn on-line, or a teacher is not available to furnish error signals or target outputs

The framework of reinforcement leaning is shown in Fig.1. And basic reinforcement learning procedure is stated as follows,

Basic learning procedure

1. step1:An agent decides an action according to observed condition in time t, and performs selected actiona_t.
2. step2: Environment is changed from s_t to s_{t+1}, and an agent acquires reward r_t corresponded with transition environment.
3. step3: $t \leftarrow t + 1$, and goto step 1.

An agent selects an action for the purpose of the maximum reward acquisition.

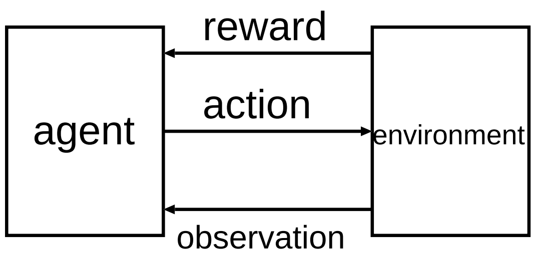

Figure 1. The framework of reinforcement learning

2.2. Q-Learning

Q-Learning is a typical reinforcement learning algorithm proposed by Watkins[6]. The learning step of Q-Learning is stated as follows,

- Step1: An agent observes its current state s.
- Step2: An agent decide an action according to suitable action selection rule such as roulette selection or ϵ-greedy method.
- Step3: An agent gets reward r from environment.
- Step4: An agent observes subsequent state s'.
- Step5: Q value is updated using next equation.

  $$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')], \quad (1)$$

  where α is a learning factor $(0 < \alpha \leq 1)$, γ is a discount rate $(0 \leq \gamma < 1)$.
- Step6: $t \leftarrow t + 1$ and goto Step1

3. Related work

3.1. Q-Routing

Q-Routing which uses the Q-Learning framework is proposed by Boyan[2]. In Q-Routing, each node x have a routing table of values $Q_x(d,y)$ for a neighbor y and destination d, of how long it takes for a packet to be delivered to node d via neighbor node y. When the node has to make a routing decision it simply chooses the neighbor y for which $Q_x(d,y)$ is minimum.

Value Q_x is updated when message is sent from node x to y using following equations.

$$t = \min_{z \in N(y)} Q_y(d,z) \quad (2)$$

$$Q_x(d,y) = \alpha(q_y + s_y + t) + (1-\alpha)Q_x(d,y), \quad (3)$$

where $N(y)$ denotes the set of neighbors of node y, d denotes source node, q_y denotes hold time within queue of node y, s_y denotes trip time from node x to y, α denotes a "learning rate" parameter.

Q-Routing can acquire better routing than an algorithm based on shortest paths on heavy traffic environment[2].

3.2. Dual Reinforcement Q-Routing

Dual reinforcement learning was developed for adaptive signal predistorters in satellite communications[4]. The same idea is used to incorporate backward exploration into the Q-Routing algorithm, named Dual reinforcement Q-Routing (DRQ routing)[3].

When a node x sends a packet to one of its neighbors y, the packet can take along same Q value information of node x. When node y receive this packet, it can make use of this information for updating its own estimate pretraining to the neighbor x. Later when node y has to make a decision, it has the updated Q value for x. The only exploration overhead is a slight increase in the size of the packets.

Q value is updated by the following equations,

$$t = \min_{z \in N(y)} Q_x(s,w) \quad (4)$$

$$Q_y(s,x) = \beta(q_y + s_y + t) + (1-\beta)Q_y(s,x) \quad (5)$$

Q_y denotes updated Q value on node y using a packet from node x to node y. Other notations are as same as Q-Routing. Using DRQ-Routing, double converge speed can be acquired theoretically.

3.3. Ants-Routing

Ants-Routing is proposed by Subramanian[5]. Ants-Routing has following two features, (1) In Q-Routing and

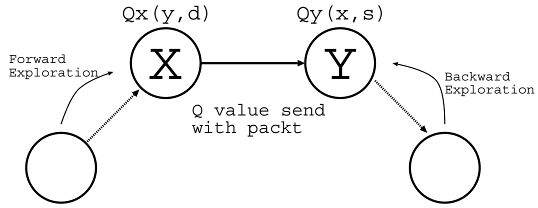

Figure 2. Forward and Backward Exploration

DRQ-Routing, estimated remain time to the distnation node is hold as Q value in routing table. In Ants-Routing, routing rules are treated as random variables. (2) Only backward exploration in DRQ-Routing is used for updating routing table.

When node r receives a messages to node x, this packet is forwarded to its neighbor y_i with probability p_i. The probabilistic table is a mechanism for exploring alternate paths in the network and keeping estimates of their lengths relatives to the current best paths.

The probabilistic table is updated using the following equations,

$$\Delta p = k/f(c) \tag{6}$$

$$P_y(s,x) = \frac{P_y(s,x)+\Delta p}{1+\Delta p} \tag{7}$$

$$P_y(s,z) = \frac{P_y(s,z)}{1+\Delta p} \quad (z \neq x), \tag{8}$$

$P_y(s,x)$ denotes sending probability from node x to node y, s is the source node, z is neighbor node of y, k is learning rate and $f(c)$ is a non-decreasing function of c. c is elapse time from the message is generated.

4. Accelerated Ants-Routing

Features of ad hoc network are frequent and unpredictable changes in network topology, as a result converging speed of routing table most important factor to evaluate routing algorithm.

In this section, accelerated Ants-Routing is proposed to accelerate converging speed. Proposed algorithm consist of two idea, therefore the probabilistic table updated method and other curtailed algorithms are as same as Ants-Routing.

4.1. No return rule

In conventional reinforcement routing, next node is selected randomly while Q value (Q and DRQ-Routing) or probability (Ants-Routing) is not converged. As a result of this selection method, packet will be back just before an adjacent node. This packet routing was unprofitable under converging rule to search an optimum path.

First idea is to eliminate "return rule" while selecting next node. This technique can be expected as the efficiency improvement of the learning at the initial stage. In reinforcement learning, ineffective rule is difficult to suppress[7]. This idea is to eliminate a detour (two step loop routing). These rule in detour may not contribute to the acquisition of the reward.

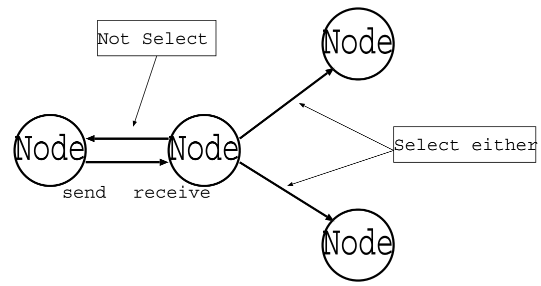

Figure 3. No return rule

4.2. N step backward exploration

In Ants-Routing, only single routing entry (one entry implies routing information for one destination node) is updated. If a packet contains own routing history, some routing entry can be updated simultaneously when single packet has been received.

Schematic diagram of this idea is shown Fig. 4. In conventional Ants-Routing algorithm, only one routing table entry toward "A" is updated (rule: a packet to node "A" will be sent toward node "C") using single packet which is sent from node "A", via node "B","C" and destination "D". On the other hand, entries toward "B" and "C" (rule: a packet to nodes "A", "B" and "C" will be sent toward node "C") are updated using proposed algorithm which contains previous two step routing informations.

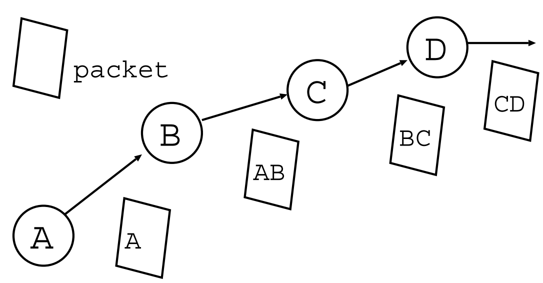

Figure 4. N step backward exploration (N=2)

The routing probability P_y on node y is updated using the following equation when a packet from x to y is arrived.

for all $m \in M$

$$\Delta p = k/f(c) \quad (9)$$

$$P_y(m,x) = \frac{P_y(m,x)+\Delta p}{1+\Delta p} \quad (10)$$

$$P_y(m,z) = \frac{P_y(m,z)}{1+\Delta p} \quad (z \neq x), \quad (11)$$

where M is a set of nodes in received packet, other parameters are same as conventional Ants-Routing.

5. Experimental results

5.1. Conditions of simulations

In this section, we present the result of extensive simulations that were performed to evaluate the proposed algorithms, and to compare their behavior with Q-Routing and DRQ-Routing.

On single simulation cycle, a node can be executed by one of following steps,

- Receive packet (if a packet to my node is in a queue)
- Relay packet (if a packet to other node is in a queue)
- Generate packet (generation rate is 5% only if no message is in a queue)
- No operation (others)

In this experiment, packet generation rate is 5% only if no other packet is in a local queue, and source and destination nodes are selected randomly. In proposed method "N step backward exploration", the value of N was set to 2 or 5.

5.2. Topology and dynamics of networks

Dynamic of the network uses in our simulations is shown in Fig. 5. Ten mobile nodes are located randomly on 8×8 grid, all links are of equal, unit cost (one unit). Each node can be transmitted within three blocks.

A packet not always reaches on dynamic network environment which topology is changed all the time. Generally, it is impossible to detect a reachability of packet using local information on a node in the framework of distributed environment.

On the other hand, our goal is accelerating converging speed, then "reachability of packet" and "converging speed of routing table" are considered in the different frameworks. Accordingly, four fixed node are located to ensure a reachability of packet at hatched node in Fig.5. By locating these fixed nodes, all node can be certainly communicated with other all node. Fixed node has the same features equal to mobile node except for a movement.

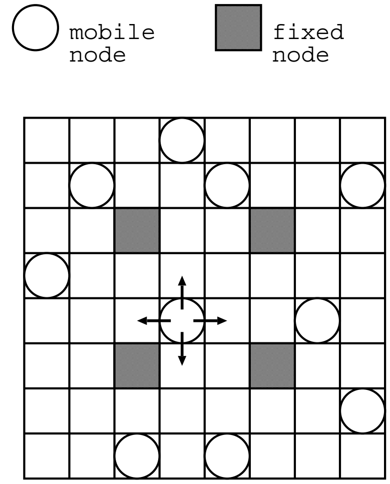

Figure 5. The field which places to the fixed node and mobile node

One mobile node which is selected randomly is moved one grid (north, east, west, south). To examine an ability of routing on dynamic topological environment, the parameter of mobilization occurrence rate MO is changed from 100% to 0%. $MO = 100\%$ means mobilization is occurred every one simulation step, and 0% means no mobilization is occurred. Furthermore, mobile node can not go to on the outside of the grid.

To evaluate the performance of the proposed algorithms, we did simulations on "Q-Routing","DRQ-Routing", "Ants-Routing", "No reverse", "N step backward exploration ($N = 2$ and $N = 5$)" and "Accelerated Ants-routing (No reverse and N step backward exploration).

5.3. Result

Our first experiment quantifies end-to-end delivery latency. Fig. 6 shows a relation between simulation steps (X axis) and average communication latency from source node to destination node (Y axis) changing mobilization occurrence rate MO (0%, 20%, 40%, 60%,80% and 100%). Unit of latency is described as simulation steps (not hops) from source node to destination node.

The plots are averages over 6 simulations. The X axis was plotted as a LOG scale at Q and DRQ-Routings (Fig.6(a) and (b)), because these two algorithms need a lot of steps for converging. Also scale of the Y axis at Q and DRQ-Routing are different from others.

At $MO = 0\%$ which means stable topology network, every routing algorithm are converged and converged result are fairly good. However Q and DRQ-routing need a lot of steps to converged. Increasing changing mobilization occurrence rate MO, converged result becomes worse. Using Q and DRQ-Routings, converged step came to grief on dynamic networks even if $MO = 20\%$.

Accelerated Ants-Routing is the best routing algorithm concerning latency and convergence on dynamic networks. N step backward exploration ($N = 5$) takes long latency compared with $N = 2$. Increasing steps for backward exploration is effective for increasing converging speed, however environmental adaptation ability becomes lower. This is because the lack of consistency between topological information within a packet and actual network topology.

Second experiment quantifies communication capacity, which describes how much packet can be received within a certain time unit. Fig.7 shows a relation between simulation steps (X axis) and number of received packet on destination node (Y axis). The gradient of this graph shows the communication capacity of this network. Using Q and DRQ-Routings, converged step came to grief on dynamic networks, then communication capacity is worse than the others. Ants-Routing has good communication capacity in stable network($MO = 0\%$). But increasing MO, communication capacity becomes worse. Proposed accelerated Ants-Routing (N=2) got good communication capacity compared with conventional Ants-Routing.

6. Conclusion

In this paper a new adaptive network routing algorithm named accelerated Ants-Routing for dynamic topology network like ad hoc network was proposed. No reverse and N step backward exploration extension achieves good acceleration for routing table convergence of conventional Ants-Routing, Q routing and DRQ-Routing even if network topology was dynamically changed. Future research will address to compare other reinforcement learning based routing algorithms such as AntNet[8], especially on their overhead.

References

[1] Tanenbaum,A.:”Computer Networks second edition”,Prentice Hall, 1989.

[2] Justin A. Boyan and Michael L. Littman:”Packet routing in dynamically changing networks: A reinforcement learning approach”, Advances in Neural Information Processing Systems, vol.6, pp 671–678, 1993.

[3] Shailesh Kumar and Risto Miikkulainen,”Dual Reinforcement Q-Routing: An On-line adaptive routing algorithm”, Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE-97, St. Louis, MO), vol.7, pp. 231-238, 1997.

[4] Patrick Goetz, Shailesh Kumar and Risto Miikkulainen :”On-Line Adaptation of a Signal Predistorter though Dual Reinforcement Learning”, Proc. 13th Annual Conf. on Machine Learning, pp.175-181, 1996

[5] Devika Subramanian, Peter Druschel and Johnny Chen : “Ants and reinforcement learning : A case study in routing in dynamic network”, In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-97), pp.832-839, 1997.

[6] J.C.H Watkins and P.Dayan: ”Q-learning”, Machine Learning Vol.8, pp.279-292,1992.

[7] Singo Kato and Hiroshi Matsuo: “A Theory of profit sharing in dynamic environment”, 6th Pacific Rim International Conference on Artificial Intelligence(PRICAI2000),Lecture Note in Artificial Intelligence 1886. pp.115-124 (2000)

[8] Di Caro, G. and Dorigo, M. ”AntNet: Distributed Stigmergetic Control for Communications Networks”, Journal of Artificial Intelligence Research, Vol. 9, pp.317-365 (1998)

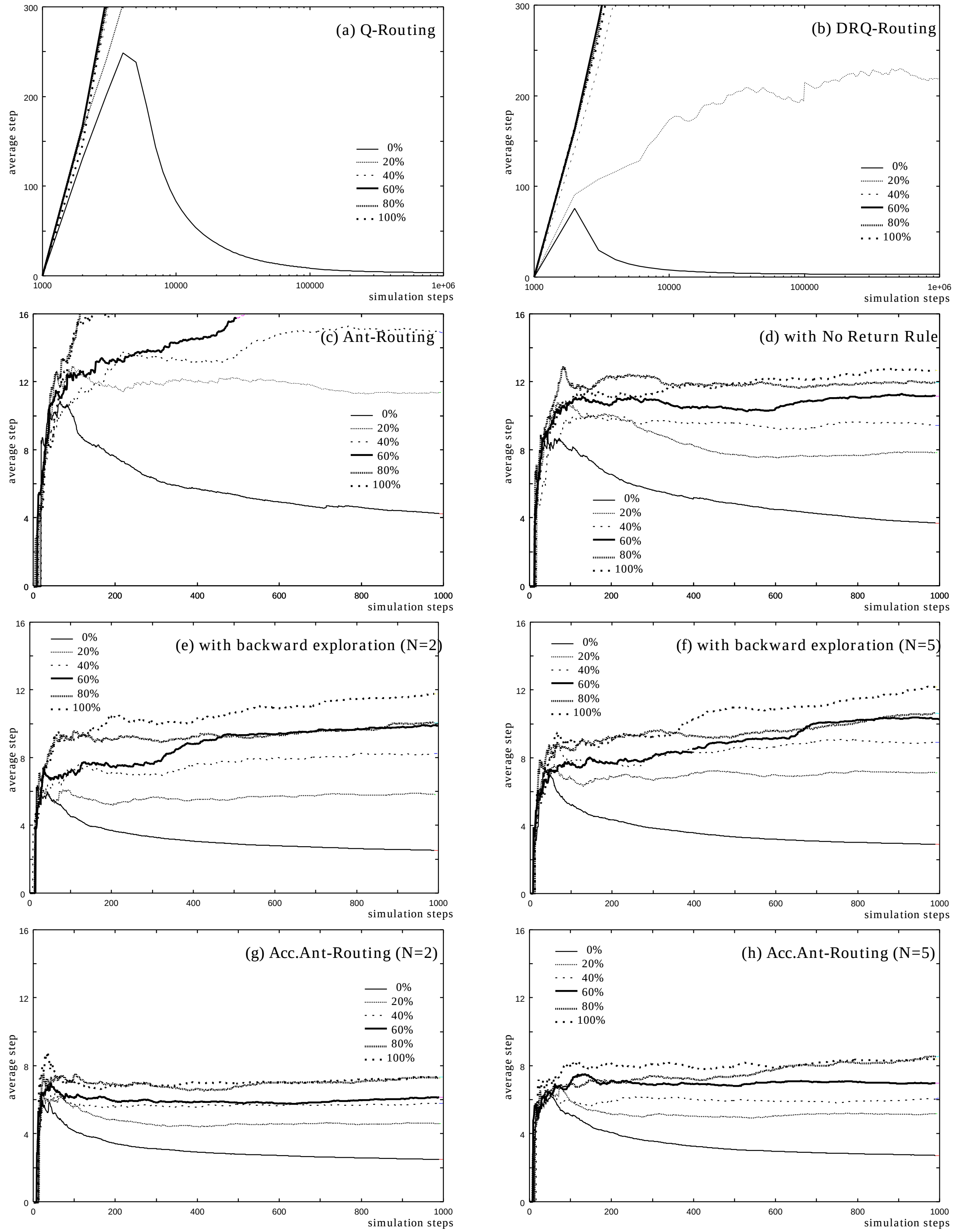

Figure 6. End-to-end delivery latency: (a) Q-Routing, (b) DRQ-Routing, (c) Ants-Routing, (d) Ants-Routing with No return rule, (e)Ants-Routing with N step backward exploration (N=2), (f) Ants-Routing with N step backward exploration (N=5), (g) Accelerated Ants-Routing (N=2) and (h) Accelerated Ants-Routing (N=5)

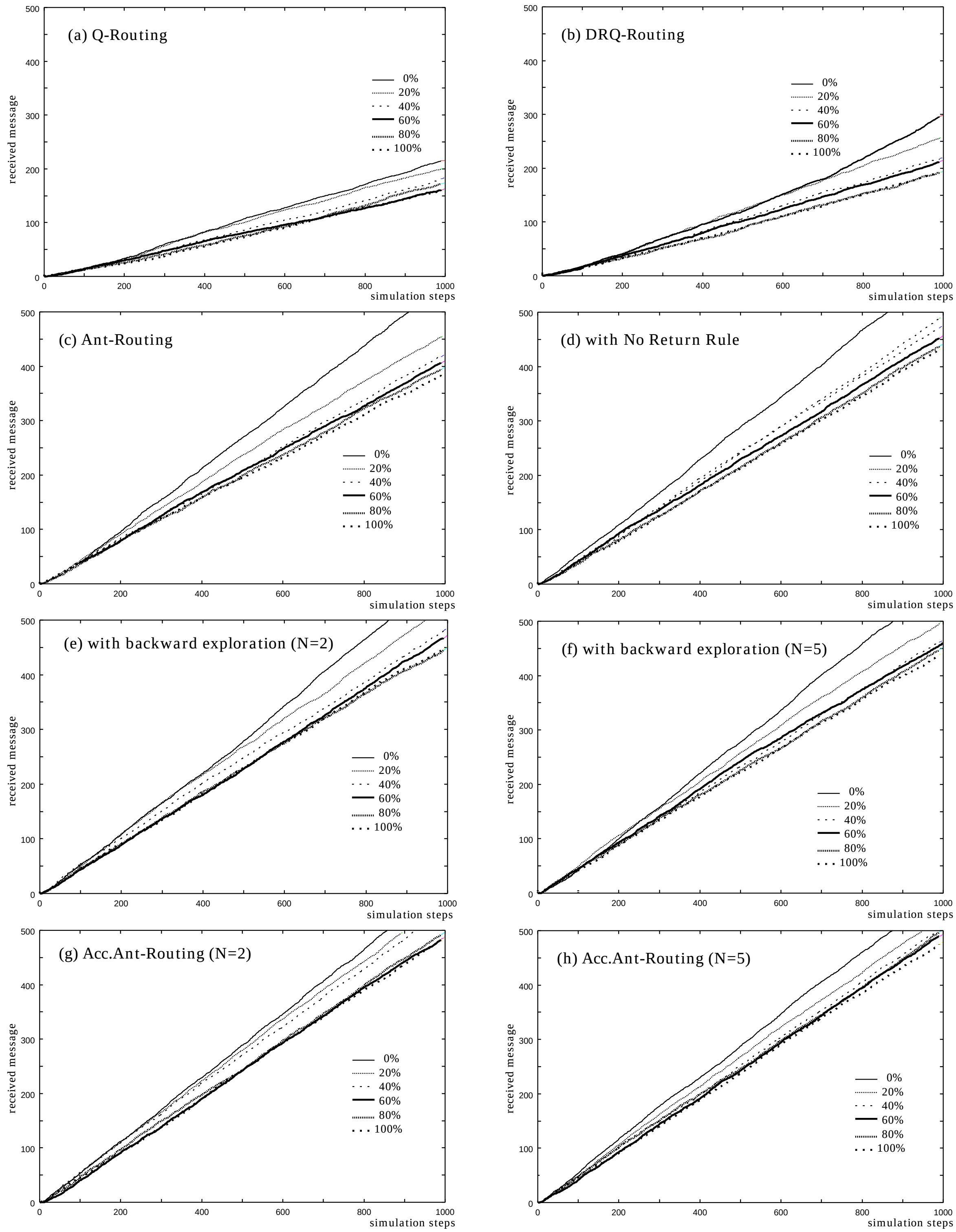

Figure 7. Communication capacity: (a) Q-Routing, (b) DRQ-Routing, (c) Ants-Routing, (d) Ants-Routing with No return rule, (e)Ants-Routing with N step backward exploration (N=2), (f) Ants-Routing with N step backward exploration (N=5), (g) Accelerated Ants-Routing (N=2) and (h) Accelerated Ants-Routing (N=5)