1A2-01

強化学習に基づいたルーティングアルゴリズムの一手法

Efficient Algorithms based on the Reinforcement Learning for Network Routing

藪内 佳孝　　加藤 昇平　　犬塚 信博
Yoshinori Yabuuchi　Shohei Kato　Nobuhiro Inuzuka

名古屋工業大学 大学院工学研究科 情報工学専攻
Department of Compurter Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

This paper proposes three efficient algorithms for network routing with the traffic changing dynamically. The paper firstly considers Q-routing, a packet routing algorithm based on the framework of reinforcement learning, and discusses the problem of the algorithm in the estimation of the cost of the routing. We, then, propose three efficient algorithms, SQ-Routing, EQ-Routing and SEQ-Routing. The algorithms make improvements of Q-Routing in both the adaptiveness and the efficiency. In this paper, we also report some experiments using a 6x6 irregular grid network with various loads and good performance results: both adaptation and efficiency.

1. はじめに

近年，未知なる環境に対して学習を行う強化学習 (reinforcement learning) に関する研究が盛んである．強化学習は，ロボットの制御やゲーム戦略の獲得などに広く応用されている．本研究では，強化学習をネットワークにおけるパケットルーティングへ応用する．強化学習の代表的アルゴリズムである Q-Learning[Watkins 92] の枠組みに基づいたルーティングアルゴリズムとして Q-Routing[Littman 93] がある．ディスタンスベクタ型ルーティングアルゴリズム [Bellman 58] にはない特徴として，Q-Routing は, 動的に変化するネットワークのトラフィックに適応してルーティング政策を変化させる．これにより，経路選択において，パケットが混雑する経路を回避する．本研究では，Q-Routing を基にネットワークへの適応能力を改善するルーティングアルゴリズムを提案する．

2. ルーティング問題

ここでは，本稿で扱うルーティング問題について説明する．ネットワークはリンクで繋がれた複数のノードから構成される．データを収めたパケットを正しい目的地に届くように経路選択を行うことをルーティングと呼ぶ．パケットはソースノードで発生し，パケットの目的地をデスティネーションノードと呼ぶ．各ノードは待ち行列を 1 つ持つ．各ノードは受け取ったパケットを一旦待ち行列に入れ，待ち行列の先頭からパケットを 1 つずつ取り出し処理を行う．パケットが次のノードへ移動することをホップと呼ぶ．ソースノードからデスティネーションノードに到着するまでに経過した時間をデリバリータイムと呼び，デリバリータイムの大部分は経路上の各ノードの待ち行列における待ち時間が占める．あるノードにおけるルーティング政策は，デスティネーションノードまでのデリバリータイムを最小にする隣接ノードを選択することである．

ソースノードからデスティネーションノードまでの経路は複数存在する．その中から最も早くデスティネーションノードに届けることが可能な経路を選択することが望ましい．もし，ネットワーク上のすべてのノードの状態を知ることができるならば，最適なルーティングが容易に決定できる．しかしながら，一般的には，ネットワーク上の全ノードの状態を知ることは難しい．そのため，隣接ノード間で情報交換を行いルーティングを決定する必要がある．

連絡先: 藪内 佳孝，名古屋工業大学大学院情報工学専攻，〒 466-8555 名古屋市昭和区御器所町，電話 052-735-5625，Fax 番号 052-735-5625，yabu@ics.nitech.ac.jp

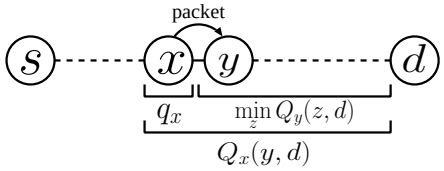

図 1: Q-Routing における $Q_x(y,d)$

3. Q-Routing

Q-Routing は，Q-Learning の枠組みに基づいたルーティングアルゴリズムである．Q-Routing では，各ノード x は隣接ノード y とデスティネーションノード d の組に対して $Q_x(y,d)$ を持ち，それらのテーブルを用いてルーティングの決定を行う．$Q_x(y,d)$ は x において，デスティネーションノードが d であるパケット P^d を x の隣接ノード y に送信したときに P^d が d に到着するまでにかかる時間の見積りである．ただし，$Q_x(y,d)$ には x の待ち行列における待ち時間を含む．つまり，P^d が x に到着してから d に到着するまでの時間である（図 1）．P^d のホップ先である隣接ノードの選択では単に $Q_x(y,d)$ が最小となるノードが選ばれる．P^d が x から y へホップしたとき，即座に y から d までの送信時間の見積りの最小値 t が x へ返される．

$$t = \min_{z \in \text{neighbors of } y} Q_y(z,d) \tag{1}$$

P^d の x における待ち時間を q_x とすると，x は次のように $Q_x(y,d)$ を更新する．

$$Q_x(y,d) \leftarrow Q_x(y,d) + \eta(q_x + t - Q_x(y,d)) \tag{2}$$

ここで，η は学習率 $(0 < \eta \leq 1)$ である．

Q-Routing では，ネットワークを構成するノードをエージェントと見なした分散強化学習アルゴリズムであり，目的（デスティネーションノード）の異なる複数のパケットが環境内に存在する．同じ目的を持つパケットを 1 つのグループと考えると，あるグループの行動（パケットをあるノードへ送信すること）は，他のグループには予測できない．そのため，環境は非マルコフ性を示す．その一方で，目的が異なるパケットが混在

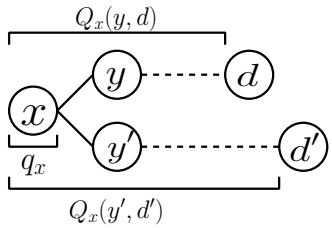

図 2: $Q_x(y,d)$ と $Q_x(y',d')$ との共通部分

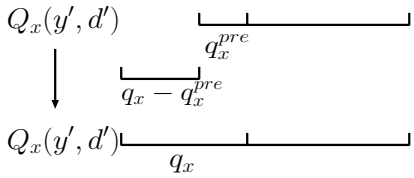

図 3: $Q_x(y',d')$ の更新

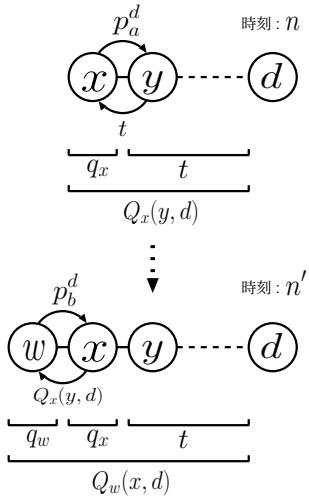

図 4: Q-Routing

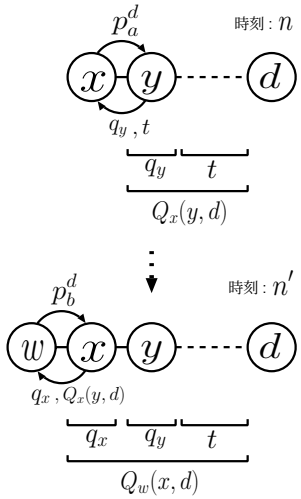

図 5: EQ-Routing

する環境であることから，Q-Routing は，ある目的のための政策の学習を他の目的のための政策の学習へも反映できる要素を持つ．

4. 提案手法

本稿では，Q-Routing の学習速度とトラフィックへの適応を改善するための 3 つのルーティングアルゴリズム SQ-Routing, EQ-Routing 及び SEQ-Routing を提案する．SQ-Routing は，各 Q 値に含まれる自身のノードにおける待ち時間がホップ先ノードやデスティネーションノードに関わらず利用できる点に着目し，Q 値の更新時にそのノードにおけるすべての Q 値に対してその待ち時間を反映させることによりトラフィックへの適応を改善する．EQ-Routing は，Q 値の新しい定義を提案することでトラフィックへの適応を改善する．SEQ-Routing は，SQ-Routing におけるアイデアを EQ-Routing に対して用いることでさらに改善する方法である．

4.1 SQ-Routing

Q-Routing では，ノード x の $Q_x(y,d)$ に x における待ち時間を含む（図 2）．$Q_x(y,d)$ の更新では，x がパケット P^d を y へホップした時点で行われるため，x における各 Q 値の更新時刻は異なる．このため，各 Q 値に含まれる x における待ち時間の見積りは異なる．しかし，x における待ち時間は x からパケットがホップするたびに q_x として更新される．したがって，前回の更新時における x での待ち時間の値を q_x^{pre} として保存しておき q_x と q_x^{pre} との差を取り Q 値に加えることでホップ先やデスティネーションノードに関わらず，すべての Q 値に含まれる待ち時間の見積りを更新できる（図 3）．

このアイデアに基づいた SQ-Routing を提案する．パケット P^d が x から y へホップしたとき，即座に y から d までの送信時間の見積りの最小値 t が x に返される．P^d の x における待ち時間を q_x とすると，x は式 (2) によって $Q_x(y,d)$ を更新し，$y' \neq y$ または $d' \neq d$ となるすべての $Q_x(y',d')$ を次のように更新する．

$$Q_x(y',d') \leftarrow Q_x(y',d') + q_x - q_x^{pre} \tag{3}$$

そして，q_x^{pre} を更新する．

$$q_x^{pre} \leftarrow q_x \tag{4}$$

Q-Routing では，x から y へ P^d がホップしたとき，$Q_x(y,d)$ のみが更新されたが，SQ-Routing では，それに加えて x におけるその他の Q 値に対して，x における待ち時間を更新することができる．これにより，x からパケットがホップする毎にホップ先やデスティネーションノードに関わらず x におけるすべての Q 値に現在の x における待ち時間が反映される．

4.2 EQ-Routing

ここでは，まず，Q 値に含まれる隣接ノードでの待ち時間について考える．ある時刻 n においてデスティネーションノードを d とするパケット P_a^d がノード x からノード y にホップした場合（図 4 上）を考える．簡単のため学習率を $\eta = 1$ とする．P_a^d の x における待ち時間が q_x で，y から x へ $t = min_z Q_y(z,d)$ が送信されたとする．このとき，x において $Q_x(y,d) = q_x + t$ である．次に，時刻 $n'(> n)$（図 4 下）においてデスティネーションノードが P_a^d と同一のパケット P_b^d がノード w から x にホップしたとする．ここで，P_b^d の w での待ち時間が q_w で，x において $Q_x(y,d)$ が最小値であった場合，$Q_w(x,d) = q_w + q_x + t$ である．このとき，$Q_w(x,d)$ の見積りに含まれる x における待ち時間 q_x の値は時刻 n' ではなく時刻 n における値である．そのため，n から n' までの間にネットワークのトラフィックが変化した場合には，時刻 n' での $Q_w(x,d)$ の信頼性が損われ有効なルーティングは期待できない．また，ノード x においてパケットの次の行き先を決める際に x における待ち時間を含めた Q 値を用いる必要はない．行き先を決める際に必要となるのは，x を出てから d に着くまでにかかる時間の見積りである．

これらを改善した EQ-Routing を提案する．$Q_x(y,d)$ の定義を次のように変更する．Q-Routing では q_x が $Q_x(y,d)$ に含まれていた（図 1）のに対し，EQ-Routing ではノード x における $Q_x(y,d)$ を x を出てから d に到着するまでにかかる時間の見積りとする（図 6）．パケット P^d が x から y へホップしたとき，即座に y から送信時間の見積りの最小値 t とその時点での y の待ち時間 q_y が x へ返される．このとき，x は次

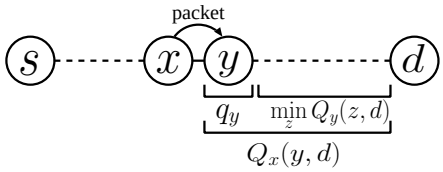

図 6: EQ-Routing における $Q_x(y,d)$

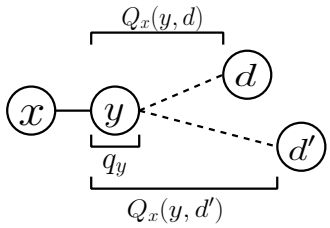

図 7: $Q_x(y,d)$ と $Q_x(y,d')$ の共通部分

のように $Q_x(y,d)$ を更新する．

$$Q_x(y,d) \leftarrow Q_x(y,d) + (q_y + t - Q_x(y,d)) \quad (5)$$

ここで，図 4 の場合と同様に EQ-Routing における Q 値に含まれる隣接ノードでの待ち時間について考える．時刻 n において P_a^d が x から y にホップし（図 5 上），y から x へ y における待ち時間 q_y と $t = min_z Q_y(z,d)$ が送信されたとする．このとき，x において $Q_x(y,d) = q_y + t$ である．次に，時刻 $n'(> n)$（図 5 下）において別のパケット P_b^d が w から x にホップしたとする．ここで，x における待ち時間が q_x で x において $Q_x(y,d)$ が最小値であった場合，$Q_w(x,d) = q_x + q_y + t$ である．このとき，$Q_w(x,d)$ の見積りに含まれる x における待ち時間 q_x の値は時刻 n' における値である．EQ-Routingでは，Q 値に含まれる隣接ノードでの待ち時間の情報が Q-Routing よりも新しく，パケットの次のホップ先となる隣接ノードを選択する際に，そのノードを出てからデスティネーションノードに着くまでの時間の見積りを用いることができる．

4.3 SEQ-Routing

SQ-Routing と EQ-Routing を組み合わせた SEQ-Routing を提案する．EQ-Routing と同様に，ノード x における $Q_x(y,d)$ を x を出てから d に到着するまでにかかる時間の見積りとする（図 6）．パケット P^d が x から y へホップしたとき，即座に y から x へ送信時間の見積り t とその時点での y における待ち時間 q_y が返される．このとき，x は $Q_x(y,d)$ を式 (5) によって更新し，$d' \neq d$ となるすべての $Q_x(y,d')$ を次のように更新する．

$$Q_x(y,d') \leftarrow Q_x(y,d') + q_y - q_y^{pre} \quad (6)$$

ここで，q_y^{pre} はこのパケットより以前に x から y へパケットがホップしたときの y における待ち時間である．$Q_x(y,d')$ 更新後 q_y^{pre} が更新される．

$$q_y^{pre} \leftarrow q_y \quad (7)$$

SEQ-Routingでは，$Q_x(y,d)$ に含まれる q_y はホップ先ノード y における待ち時間であるので，ホップ先ノードの数だけ待ち時間を記憶する．例えば，x の隣接ノードが y と z の 2 つであれば，x では q_y^{pre} と q_z^{pre} として 2 つの値を記憶しておく．

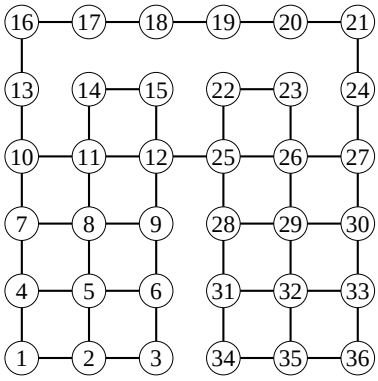

図 8: 6x6 irregular grid におけるネットワークのトポロジー

5. 評価実験

5.1 実験環境

6x6 irregular grid[Littman 93, Boyan 94] を用いて実験を行った．図 8 にネットワークのトポロジーを示す．各ノードはパケットを格納しておく待ち行列（FIFO キュー）を 1 つ持つ．1 シミュレーションタイムステップ（以降，ステップ）において各ノードは自分の待ち行列の先頭からパケットを 1 つ取り出し，それを隣接ノードへ送信する．パケットを受け取ったノードは，自身がそのパケットのデスティネーションノードである場合，そのパケットをネットワーク上から削除し, そうでない場合は待ち行列にそのパケットを加える．1 ステップ当たりに発生するパケットの数をネットワークロード（$load$）と呼ぶ．パケット発生時において，ソースノード及びデスティネーションノードは，それぞれランダムに決定される．デリバリータイムは，パケットがソースノードからデスティネーションノードへ到着するまでにかかったステップ数とする．

図 8 のネットワークの特徴は，右半分側と左半分側をつなぐリンクが 18-19 間と 12-25 間の 2 つあることである．すべてのパケットがホップ数を最小とする経路を選択すれば，12 と 25 における待ち時間が増加する．そのため，12 及び 25 における待ち時間が長い場合は，迂回しても 18-19 間を通る経路の方がデリバリータイムを短くできる．

実験では，各ノードにおける Q 値の初期値は十分小さくランダムな正数とし，学習率は $\eta = 0.7$ とした．このパラメータの下で $load$ が 0.5〜4 の場合において実験した．

5.2 実験結果

実験では，Q-Routing，及び，提案した 3 つの手法に加えて，ディスタンスベクタ型ルーティングアルゴリズムである Shortest Path との比較も行った．Shortest Path では，シミュレーションを開始する前に全てのノード間の距離（ホップ数）をあらかじめ計算しておき，パケットのホップ先は常にデスティネーションノードまでの距離が最小となる経路が選択される．

図 9，10 及び 11 に，$load = 1.0$（low load），$load = 2.5$（medium load）及び $load = 3.75$（high load）における学習中のデリバリータイムの推移をそれぞれ示す．同図において，average delivery time とは，50 ステップ毎のデリバリータイムの平均を示す．Shortest Path は low load においてはデリバリータイムが最も小さい．medium load，high load のようにネットワーク上のパケット数が多い場合では 12-25 間のトラヒックが増加しステップの経過と共にデリバリータイムが増加してしまう．一方で，強化学習を用いたルーティングアルゴリ

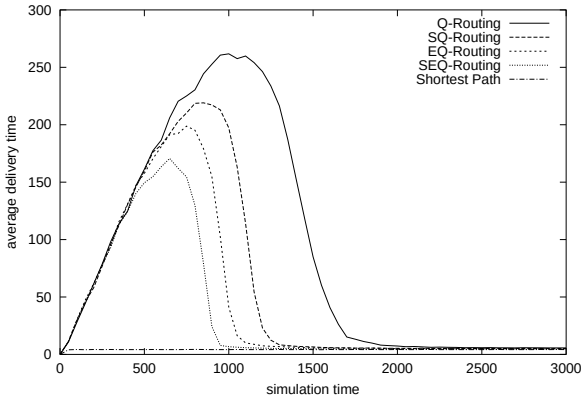

図 9: low load におけるデリバリータイムの変化

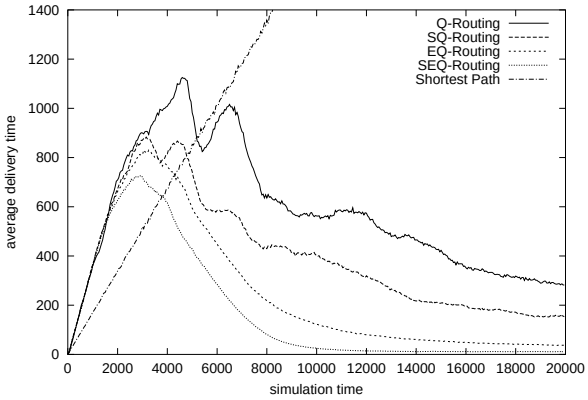

図 11: high load におけるデリバリータイムの変化

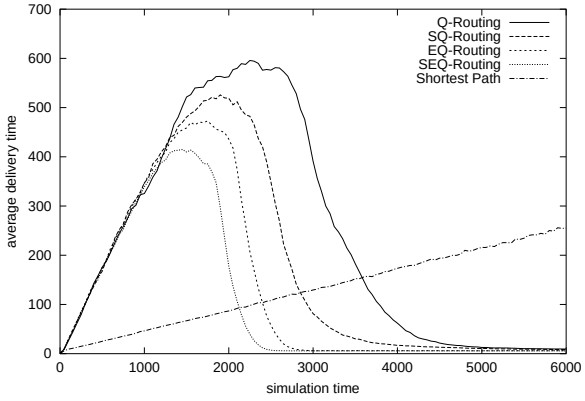

図 10: medium load におけるデリバリータイムの変化

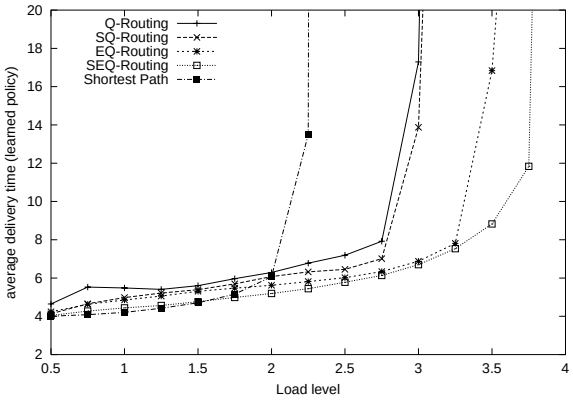

図 12: 各 load における学習後のデリバリータイム

ズムは，学習初期においてはデリバリータイムが増加するが，学習が行われるにつれ減少する．これらの結果は強化学習によるルーティングアルゴリズムが，トラフィックの増加に伴い迂回経路を適切に選択していることを示している．いずれの *load* においても SEQ-Routing，EQ-Routing，SQ-Routing，Q-Routing の順に高い効果が確認された．

図 12 に学習後における各ネットワークロードでのデリバリータイムを示す．$0 < load < 1.5$ ではわずかながら Shortest Path に劣るものの，全体的に SEQ-Routing が最もルーティング性能が良いことがわかる．

6. おわりに

本稿では，強化学習を応用したルーティングアルゴリズムである Q-Routing の改良手法として，SQ-Routing，EQ-Routing 及び SEQ-Routing を提案した．また，これらの比較実験として 6x6 irregular grid 問題を扱った．その実験において，提案手法が Q-Routing よりも学習が早く，学習後においても効率的にパケットを送信できることを確認した．

関連研究として，CQ-Routing，DRQ-Routing や CDRQ-Routing[Kumar 99] がある．これらも Q-Routing を基にしたルーティングアルゴリズムであり，本提案手法で用いた改善方法がこれらの方法に適用できる．今後の課題としてこれらのアルゴリズムへの本手法の適用が挙げられる．

謝辞

本研究に関して貴重なご助言をいただいた名古屋工業大学伊藤英則教授に深く感謝致します．

参考文献

[Bellman 58] R.Bellman. On a routing problem. Quarlerly of Applied Mathematics, 16:87-90, 1958.

[Littman 93] Michael Littman and Justin Boyan. A Distributed Reinforcement Learning Scheme for Network Routing. Technical Report CMU-CS-93-165, School of Computer Science, Carnegie Mellon University, 1993.

[Boyan 94] Justin A.Boyan and Michael L.Littman. Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach. Advances in Neural Information Processing Systems 6. MIT Press, Cambridge, MA, 1994.

[Kumar 99] Shailesh Kumar and Risto Miikkulainen. Confidence Based Dual Reinforcement Q-Routing:An adaptive online network routing algorithm. the 16 International Joint Conference on Artificial Intelligence, 1999.

[Sutton 98] Richard S.Sutton and Andrew G.Barto. Reinforcement Learning:An Introduction. MIT Press,1998.

[Watkins 92] Watkins,C.J.C.H.and Dayan,P.(1992). Q-Learning. Machine Learning, 8:279-292.