

パケットルーティングネットワークモデルにおける 効率的スケジューリング

土田 隆文[†]

浅野 孝夫[‡]

中央大学大学院 理工学研究科 情報工学専攻[†]

中央大学 理工学部 情報工学科[‡]

1 はじめに

ネットワークに投入されたパケットに対して、ルーティングとスケジューリングを効率良く行うことにより、ネットワークの安定を保つことを目的とする問題を扱う。ここで、パケットの待ち時間を最小化することにより「安定」を実現するものとする。このモデルにおいて、通常よりも厳しい、すなわち、結果が悪くなるような入力をあえて与えたときでも、安定した結果を出力するアルゴリズムの開発を考えている [2]。本稿では、この問題に対する従来のアルゴリズムと、さらにスケジューリングの部分に改善を加えたものを実装する。また、それによって得られた結果に基づいて、実験的性能評価を行う。

2 問題定義

有向グラフ $G = (V, E)$ にパケットを投入する。本問題では、点 $v \in V$ をノード、枝 $e \in E$ をリンクと呼ぶものとし、ノード数を n 、リンク数を m で表す。このとき、すべてのリンクには、混雑関数 $c(e)$ が割り当てられている。本稿では、パケットを一時的に蓄えるバッファ P と、出力を待つパケットを蓄えるキュー Q を、ノードが持つタイプをノードシステム、リンクが持つタイプをリンクシステムと呼ぶことにする[§]。文献 [1] ではリンクシステムを扱っているが、本稿ではノードシステムに対しても実験的性能評価を行う。また、このキューは、単位時間に単位パケットしか出力できないものとする。

各パケットはソースノード、シンクノードを持ち、ソースノードに発生した時点で、シンクノードまでルーティングされる。また、パケットがシンクノードに到達するまでに、ルート上の各キューに留まり、出力を待った時間 (ステップ) の合計を待ち時間と呼ぶ。

ここで、本問題の目的は、ルーティングとスケジューリングを効率良く行うことにより、すべてのパケットの待ち時間のうち、最大のを最小化し、ネットワークの安定を保つことである。

A Study of Effective Scheduling in Packet Routing Network Model

[†]Takafumi TSUCHIDA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

[‡]Takao ASANO, Department of Information and System Engineering, Faculty of Science and Engineering, CHUO University

[§]本稿の説明は、リンクシステムとなっている。

2.1 時間の流れ

本問題では、時間はフェイズと呼ばれる離散ステップで進む[¶]。このとき時間を、 w 個のステップからなる t 個のタイムウィンドウ (以下ウィンドウ) に分ける。すなわち、各フェイズは t 個のウィンドウから、各ウィンドウは w 個のステップからなる。

2.2 パケットの投入

パケットはあるウィンドウ中に、割合 r (≤ 1) で投入される。すなわち、各ウィンドウ中に $\lfloor wr \rfloor$ 個ずつのパケットが、それぞれのソースノードに投入される。また、前述のとおり、パケットは発生した時点でルーティングされるが、すぐにはネットワークには投入されない。ルーティングが済んだパケットは発生した次のウィンドウの最初にネットワークに投入される。具体的には、時刻 $[0, w)$ の区間に発生したパケットは、時刻 w にネットワークに $\lfloor wr \rfloor$ 個まとめて投入され、それ以降スケジューリングされるということになる。そして、このパケット投入方法が通常よりも厳しい、すなわち、結果が悪くなるような入力ということになる。

3 Andrews らのアルゴリズム [1]

3.1 ルーティング

ネットワークに発生したパケットに対し、混雑関数 c のもとでの最短パスを探し、選ばれたリンクの混雑を更新する。すなわち、ルーティングアルゴリズムは以下のように書ける。

```

全リンクの混雑関数を初期化 :  $c(e) = \delta^t, \forall e$ 
for  $i$  ( $= 1, \dots, t$ ) 番目のウィンドウ
  for  $i$  番目のウィンドウ中に発生した各パケット  $j$ 
     $p := c$  のもとでの  $j$  のソース・シンク間の最短パス
     $p$  に選ばれた各リンク  $e$  の混雑関数を更新 :
       $c(e) := c(e)(1 + \mu/w), \forall e \in p$ 
    
```

3.2 スケジューリング

各パケット j には、与えられたパス上のすべてのリンク e に対するデッドライン τ_e^j を与え、それを優先度

[¶]ここで、任意の正の整数 w 、パケットの投入率 r (≤ 1) に対して、 μ は r に、 δ 、 t は r 、 m に依存したパラメータとする。

とする。デッドラインの小さいパケットから出力し、次のリンクに進んだパケットのデッドラインを更新する*。

```

for  $i (= 1, \dots, t)$  番目のウィンドウの各ステップ
  for  $\forall e$ 
    for  $i$  番目のウィンドウ中に投入された各パケット  $j$ 
       $[iw + T, (i + 1)w - d_{\max}T]$  から、
      一様ランダムにデッドライン  $\tau_e^j$  の初期値を選ぶ
       $P_e$  (リンク  $e$  のバッファ) に  $j$  を格納する
    for 新たにリンク  $e$  に到着した各パケット  $j$ 
      デッドラインを更新する:  $\tau_{e+1}^j = \tau_e^j + T$ 
       $P_e$  に  $j$  を格納する
    for  $Q_e$  (リンク  $e$  のキュー) 内の各パケット  $j$ 
      デッドラインの最も小さいパケットを出力する
    for  $Q_e$  に残った各パケット  $j$ 
      待ち時間を 1 増やす
       $P_e$  内のパケットを  $Q_e$  に格納する
  
```

4 提案手法

Andrews らのアルゴリズムの問題点

以上の Andrews らのスケジューリングアルゴリズムには 2 つの問題点があると考えられる。1 つ目として、残ホップ数が 1 のパケットと、残ホップ数が多いパケットの待ち時間が無駄に増加してしまうという点、2 つ目として、デッドラインの初期値を一様乱数で与えているため、ルーティングが済んでいるという利点が活用できていないという点である。これらの問題点を解消するために、以下の手法を適用する。

提案手法

手法 I 各ステップ、各ノードにおいて、残ホップ数が 1 のパケットのデッドラインを T 減らす。

手法 II 各ステップ、各ノードにおいて、残ホップ数が最も多いパケットのデッドラインを T 減らす。

手法 III デッドラインの初期値の選び方を以下のように変更する。発生したパケット j がルーティングされた時点で、 j のシンクまでのホップ数 d^j が、 d_{\max} の 0.4 倍以上であれば、 $[iw + T, Md]$ から、そうでなければ、 $[Md, (i + 1)w - d_{\max}T]$ からデッドラインの初期値を選ぶ^{||}。

5 実験的性能評価と考察

a Andrews らのアルゴリズム (ノードシステム)

b a に手法 I, II を適用したもの

*ここで、 $e + 1$ は、パケット j が向かう次のリンクを表すものとし、 T は w と m に依存したパラメータとする。また、 d_{\max} は、ソースからシンクまでのホップ数のうち、その時点までの最大のものであり、このとき、 $iw + T < (i + 1)w - d_{\max}T$ となる。

^{||}Md は、 $[iw + T, (i + 1)w - d_{\max}T]$ の中間値とする。

c Andrews らのアルゴリズム (リンクシステム)

d c のスケジューリングの部分に手法 I, II を適用したもの

e d に III を適用したもの

以上のアルゴリズムで、各リンク数、ステップ数、ノード数に対して 100 個のランダムなインスタンスを解き、その平均を比較した。図 1 は、ノード数を 100、リンク数を 300 で固定し、ステップ数を変化させたとき、インスタンスを c, d, e で解いた解のうち、平均最大待ち時間の比較である。全体を通して、Andrews らのアルゴリズムによる結果が悪くなればなるほど、提案手法とのギャップが大きくなっていることがわかる。

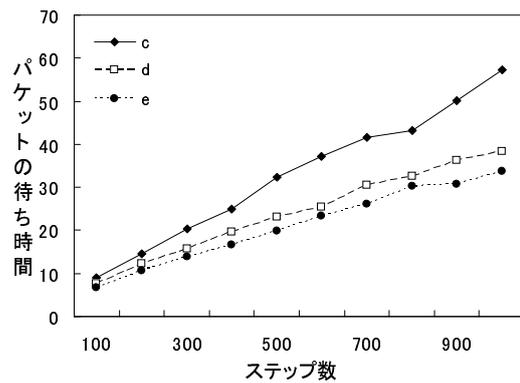


図 1 ステップ数の変化

6 むすび

本稿では、通常よりも厳しい、すなわち、結果が悪くなるような入力を与えても、ネットワークの安定を保つことのできる、Andrews らのスケジューリングアルゴリズムに対する実験的性能評価を行った。また、その安定を保ったまま結果を改善することのできる手法を提案し、それらを実験的に性能評価した。その結果、Andrews らのアルゴリズムで特に悪い結果が出力される際に、良い結果を出力することができた。さらに、バッファとキューをノードが持つ、ノードシステムに対しても実験を行い、同様の結果を得ることができた。

謝辞

本研究は、一部、文部科学省科学研究費補助金からの援助のもとで行われたものである。

参考文献

- [1] M. Andrews, A. Goel, L. Zhang and, A. Fernandez: Source Routing and Scheduling in Packet Networks. *42nd IEEE symposium on Foundations of Computer Science*, Las Vegas, Nevada, October 2001, pp. 168–177.
- [2] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson: Adversarial queueing theory. *Journal of the ACM*, January 2001, 48(1): pp. 13–38.