

輻輳通知の時間遅れを考慮したボトルネックルータ検知と 高精度フロー制御手法の提案

— 輻輳予測の有効性検証を目指して —

長田 智和† 宮里 智樹‡ 谷口 祐治§ 玉城 史朗†

† 琉球大学理工学研究科総合知能工学専攻

‡ 東京工科大学工学部情報通信工学科

§ 琉球大学総合情報処理センター

概要

近年、インターネットの普及により、GigabitEthernet、ATM技術などを使ったネットワークの高速化が進み、様々なネットワークサービスの提供によって通信トラフィックが爆発的に増加している。

このような状況において、ネットワークにおけるルータがボトルネックとなり、パケット破棄や遅延を発生させ、通信スループットを低下させるケースが多くなっている。これは、既存のTCPにおける輻輳制御の仕組みがEnd-to-Endでの制御に依存し、ネットワークの状態が輻輳制御に正確に反映されていないことが原因であると考えられる。

本研究は、実際のネットワークを想定したシミュレーションから、ネットワークの輻輳状態を把握して輻輳制御を行うことにより、高精度なフロー制御が可能であることを示し、更に、ボトルネックルータを予測することによって、パケット破棄を軽減する手法について検討する。

A Proposal of the Bottleneck Router Detection Mechanism considered Time Lag of Congestion Notification and High Precise Flow Control Technique - Aiming at the Verifficetion of Effectivity of Congestion Forcasting -

Tomokazu NAGATA† Tomoki MIYAZATO‡ Yuji TANIGUCHI§ Shiro TAMAKI†

† Department of Information Engineering, University of the Ryukyus

‡ Department of Information Networks, Tokyo University of Technology

§ Center for Integrated Information Processings, University of the Ryukyus

Abstract

In recent years, by the spread of Internet, speedup of network, using technology of GigabitEthernet and ATM, progresses. In addition, various network services is done providing, the network traffic is explosive and increases.

In such a circumstances, the router in network generates bottleneck or packet loss and delay, and there are many cases letting communication throughput deteriorate. We thought that congestion avoidance in traditional TCP depends on the control with End-to-End and the network status is not reflected precisely to flow control, is cause.

In this study, we show that possibility of high precise flow control by using congestion status of router and doing congestion avoidance from the simulation that assumed true network. Furthermore, we examine technique to reduce packet loss by predicting bottleneck router.

1 はじめに

近年、インターネットの普及により、Gigabit Ethernet, ATM 技術などを使ったネットワークの高速化が進み、様々なネットワークサービスの提供によって通信トラフィックが爆発的に増加している。

このような状況において、ネットワークにおけるルータがボトルネックとなり、パケット破棄や遅延を発生させ、通信スループットを低下させるケースが多くなっている。これは、既存の TCP における輻輳制御の仕組みが End-to-End での制御に依存し、ネットワークの状態がフロー制御に正確に反映されていないことが原因であると考えられる。

本研究は、実際のネットワークを想定したシミュレーションから、ネットワークの輻輳状態を把握して輻輳制御を行うことにより、高精度なフロー制御が可能であることを示し、更に、ボトルネックルータを予測することによって、パケット破棄を軽減する手法について検討する。パケット破棄を軽減することで再送も軽減でき、結果的にネットワーク全体の通信キャパシティを向上させることができる。

本論文では、まず、2章で既存の輻輳制御の概要と問題点を述べ、次に、3章で確認応答(ack)メッセージのウィンドウサイズ改竄による輻輳制御の性能改善の試みについて述べる。更に、4章では輻輳予測のフロー制御への応用について見通しを述べ、最後に、5章でまとめとする。

2 TCP/IP 輻輳制御

インターネット上で最も広く用いられているトランスポートプロトコル“TCP(Transmission Control Protocol)”は、パケット破棄やRTT(Round Trip Time)によってネットワークの輻輳状態を検知し、送出データグラムのウィンドウサイズを動的に更新することで輻輳制御を行っている。しかし、この仕組みは End-to-End の確認応答(ack)メッセージの交換によって行われるため、広域網のような複数のルータを経由する通信環境においては、ネットワーク上の輻輳状態がフロー制御に反映されないという問題点がある。また、ネットワークの輻輳状態をフロー制御に反映させるため、RFC3168 で示されている輻輳通知(ECN:Explicit Congestion Notification)が提案されているが、高精度な輻輳制御の実現には至っていない。

輻輳制御は大きく暗示的輻輳制御と明示的輻輳制御の2つに分類される。まず、これらの概要を以下で説明する。

2.1 暗示的輻輳制御

暗示的輻輳制御は、図1に示すような TCP が備えている輻輳制御である。TCP は End-to-End で信頼性のある通信を提供するため、パケット破棄やタイムアウト等のエラーを検出すると、再送やウィンドウサイズの更新によって動的なフロー制御を行う。特に、送信側では確認応答(ack)メッセージのタイムアウトによってネットワークの輻輳状態を“間接的”に判断することから、暗示的輻輳制御と呼ばれている。

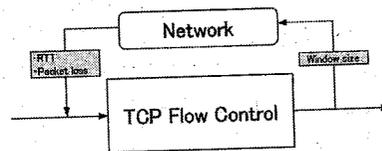


図1 : TCP 暗示的輻輳制御

図1からも分かるように、暗示的輻輳制御はパケット破棄やタイムアウト発生の情報をもとにフィードバック制御を行うが、間接的な情報に基づくため、一般に次のような問題点が指摘されている。

1. 輻輳発生時の回復対応が遅く、特に広域網では通信遅延の影響が大きい
2. 不必要なスループット低下を招く

これらの暗示的輻輳制御の問題点を改善するため、次に示す明示的輻輳制御が提案されている。

2.2 明示的輻輳制御

明示的輻輳制御は、ルータでの輻輳発生やその兆候を明示的にエンドホストへ通知し、その情報をもとにフロー制御を行う。ネットワークの輻輳状態を“直接的”に判断することから、明示的輻輳制御と呼ばれている。

明示的輻輳制御には、ERD(Early Random Drop gateway), ICMP(Internet Control Message Protocol) 始点抑制, DECbit アルゴリズム, RED(Random Early Detection) ゲートウェイ方式などが提案されている。ここでは、代表的な“RED ゲートウェイ方式”について、その概要を以下で説明する。

2.2.1 RED ゲートウェイ方式

RED ゲートウェイ方式では、輻輳通知 (ECN) ビットをデータグラムヘッダに設定する。送信側は初期状態で ECN=0 にセットしてデータグラムを送出し、ルータでは輻輳が発生していなければそのままデータグラムを転送し、輻輳が発生する (又はその兆候がある) と ECN=1 にセットしてデータグラムを転送する。受信側では、ECN=1 であるデータグラムを受け取ると、そのデータグラムに対する確認応答 (ack) メッセージの輻輳情報 (CI: Congestion Indication) ビットをセットし、送信側へ通知する。これによって送信側は輻輳を検知し、文献 3 における ECN による輻輳回避手順に従ってウィンドウサイズを更新する。更に、RED ゲートウェイ方式は、ECN ビットのセット確率によってバッファ内のデータグラムをランダムに破棄する (図 2)。

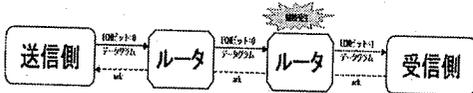


図 2 : RED ゲートウェイ方式の輻輳通知

RED ゲートウェイ方式の特徴として以下のものが挙げられる。

1. 平均バッファ長の {最小 | 最大} 閾値を適切に設定することによって、平均バッファ長の制御が可能 (バッファ溢れによるデータグラム破棄を軽減することが可能)
2. 平均バッファ長の制御が可能であることから、通信遅延の制御が可能
3. 特定コネクションから突発的なトラフィックが発生した場合、そのコネクションのデータグラムの連続的な破棄の軽減が可能

以上のように、RED ゲートウェイ方式では、ルータの平均バッファ長から輻輳状態を判断し、ECN によって送信側に輻輳通知を行うことで、ネットワークの状態をフロー制御に反映することができる。

2.3 シミュレーション (参考実験)

RED ゲートウェイ方式の動作検証のため、ネットワークシミュレータ “ns”⁽⁴⁾ を用いて、シミュレーションによる参考実験を行った。

2.3.1 RED ゲートウェイ方式の動作検証

まず、図 3 に示すネットワークモデルを作成した。

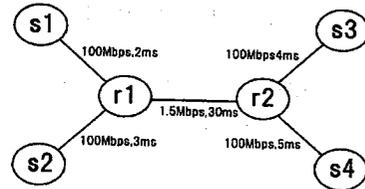


図 3 : ネットワークモデル

図 3 において、s1 - s4 はエンドホスト、r1, r2 はルータである。r1 - r2 間のバッファには RED ゲートウェイ方式を使用する。また、s1 ~ s4 では、TCP バージョン “Reno” によるフロー制御を行う。更に、RED ゲートウェイ方式では ECN ビットモードを ON とする。また、r1 - r2 間のそれぞれのバッファサイズは 32 セグメント (1 セグメント = 1024 bytes) とし、RED ゲートウェイ方式における平均バッファ長の最小閾値を 16、最大閾値を 24 とする。

ここで、s1 → s3, s2 → s3, s3 → s2, s4 → s2 の 4 つの FTP 通信 (複数フロー) を行い、ルータ (r1) におけるフローの観測を行った (図 4)。比較として、r1 - r2 間のバッファを DropTail 方式 (ECN ビットモード: OFF) の場合の観測も行う (図 5)。

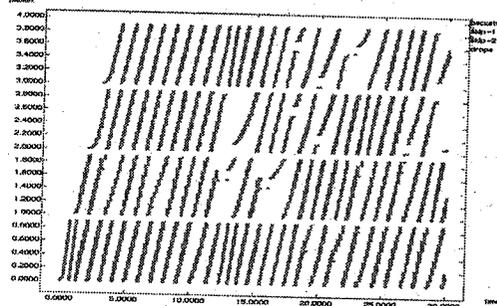


図 4 : データグラムフロー (RED 方式)

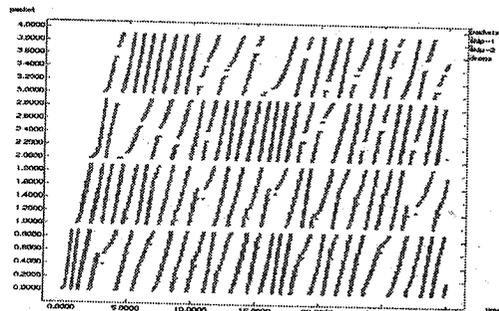


図 5 : データグラムフロー (DropTail 方式)

図4, 5から, r1-r2のボトルネックリンクにおいて, REDゲートウェイ方式がDropTail方式よりもパケット破棄が軽減されており, 結果として, 再送が軽減され, スループットが向上していることが確認された。

更に, 同じ条件で, ルータ(r1)におけるバッファ長の観測を行った(図6, 7)。

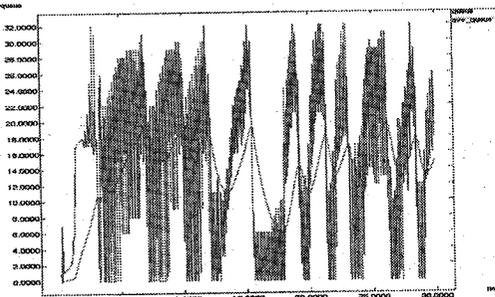


図6: バッファ長 (RED方式)

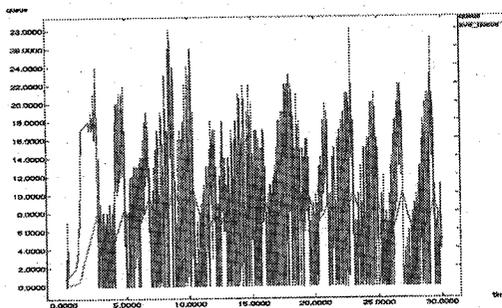


図7: バッファ長 (DropTail方式)

図6, 7から, REDゲートウェイ方式ではDropTail方式に比べて平均バッファ長が高いことが確認された。これは, REDゲートウェイ方式では平均バッファ長が最小閾値(=16)及び最大閾値(=24)の間で調整されているためだと考えられる。更に, REDゲートウェイ方式ではECNによる送信側への輻輳通知によってフロー制御も行われるため, バッファ溢れによるデータグラム破棄を軽減でき, 図4にはその効果が現れていることも確認できる。

2.3.2 REDゲートウェイ方式の問題点

前節でのシミュレーションによる参考実験でも示した通り, REDゲートウェイ方式は輻輳制御において効果的であることが確認された。

しかし, REDゲートウェイ方式では, 輻輳情報は送信データグラムにECNビットを設定し, その確認

応答(ack)メッセージによって送信側に通知するため, 輻輳情報の伝達に遅れが生じる事は明らかである。すなわち, 送信側への輻輳情報の伝達に, 最悪の場合1*RTTの遅れが生じることになる。一般にRTTは, LANにおいて数ms, WANに至っては数十ms~数百msに達する。これは, ルータでのデータ送出処理時間を考えると大きな遅れと見ることができ, 送信側が輻輳を検知したときには既にルータの状態が変化していることも十分考えられる。

そこで, 通信遅延が輻輳制御に与える影響を確認するため, 図3のネットワークモデルにおいてs1→s3のFTP通信(単一フロー)を行い, r1-r2間のリンクの物理伝送遅延が30ms, 50ms, 100msの場合について, ルータ(r1)におけるパケット破棄率の算出及び平均バッファ長の観測を行った(図8)。

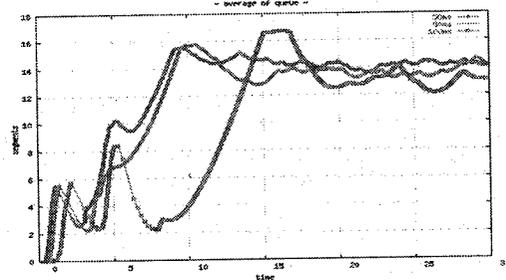


図8: 平均バッファ長 (単一フローの場合)

パケット破棄率:

0.85% (30ms), 0.98% (50ms), 3.94% (100ms)

次に, 複数フローの場合について4つのFTP通信(s1→s3, s2→s3, s3→s2, s4→s2)を行い, ルータ(r1)におけるパケット破棄率の算出及び平均バッファ長の観測を行った(図9)。

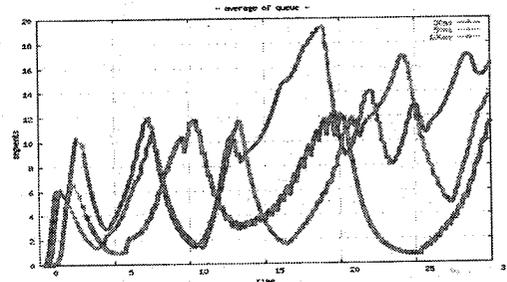


図9: 平均バッファ長 (複数フローの場合)

パケット破棄率:

0.28% (30ms), 0.81% (50ms), 4.66% (100ms)

図8, 9及びパケット破棄率より, 単一および複数フロー両方ともに, 瞬時変動の様子は異なっていないものの, 通信遅延が大きいほどパケット破棄率及び平均バッファ長とも大きくなっていることが確認できる. これより, 通信遅延によって輻輳通知に時間遅れが生じたことが輻輳制御に影響を与えていることが確認できる.

このシミュレーション結果は, 輻輳通知の時間遅れを考慮した仕組みを導入することによって, より精度の高い輻輳制御の可能性を示唆していると言える.

3 提案手法

3.1 確認応答(ack)メッセージのウィンドウサイズ改竄

RED ゲートウェイ方式の動作検証の結果を踏まえ, 輻輳通知の時間遅れを考慮したフロー制御の試作として次の手法を提案する.

提案手法: ルータにおいてデータグラムに対する確認応答(ack)メッセージのウィンドウサイズを改竄する. 具体的には, 送信側が許容可能なウィンドウサイズ(a)とルータが許容可能なウィンドウサイズ(b)を比較し, “a > b”であれば確認応答(ack)メッセージのウィンドウサイズをルータが許容可能なウィンドウサイズ(b)に改竄する. 以下同様に, 確認応答(ack)メッセージが経由する全てのルータで行う.

以上の方法によって, パス上に存在するボトルネックルータの許容可能なウィンドウサイズを送信側へ通知することができ, かつ, 最悪の場合でも $\frac{1}{2}$ RTTの時間遅れで輻輳通知を行うことができる.

3.2 シミュレーション(評価実験)

提案手法の性能評価のため, 試作モデルをns上で実装した. シミュレーションは, 図3のネットワークモデルにおいて単一フローの場合としてFTP通信(s1 → s3)を行い, r1 - r2間のリンクの物理伝送遅延が30ms, 50ms, 100msの場合について, ルータ(r1)におけるパケット破棄率の算出及び平均バッファ長の観測を行った(図10).

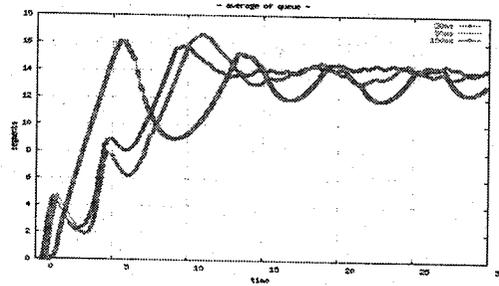


図10: 平均バッファ長(単一フローの場合)

パケット破棄率:

0.83% (30ms), 0.89% (50ms), 0.71% (100ms)

次に, 複数フローの場合として4つのFTP通信(s1 → s3, s2 → s3, s3 → s2, s4 → s2)を行い, ルータ(r1)におけるパケット破棄率の算出及び平均バッファ長の観測を行った(図11).

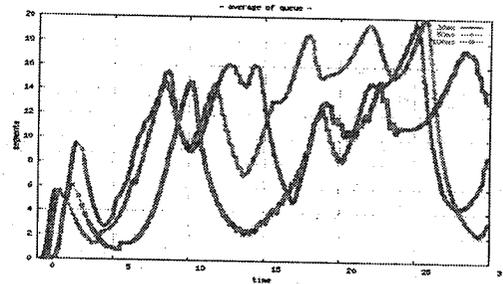


図11: 平均バッファ長(複数フローの場合)

パケット破棄率:

0.34% (30ms), 2.33% (50ms), 5.89% (100ms)

まず, 単一フローの場合では, RED ゲートウェイ方式に比べて提案手法は, 通信遅延が大きくなった場合でもパケット破棄率が低く, 輻輳制御の性能改善の効果が得られていることが確認できる.

しかし, 複数フローの場合では, 通信遅延によってはパケット破棄率がREDゲートウェイ方式に比べて悪化しているケースも発生している.

3.3 考察

提案手法において, 輻輳通知の時間遅れを小さくしたにも関わらず, 複数フロー下ではパケット破棄の軽減効果が得られなかった原因は次のように考えられる. すなわち, 複数のフローが存在した場合, ルータでのフローごとの確認応答(ack)メッセージ改竄

のタイミングが非同期であり、また、ルータとエンドホスト間の遅延も異なっているため、例えばあるフローが確認応答 (ack) メッセージのウィンドウサイズに従ってウィンドウを減少させた場合、ボトルネックルータの許容可能なウィンドウサイズが大きくなり、他のフローがその分ウィンドウを広げてしまうという現象が起こっていると考えられる。このため、個々のフローでは意図したウィンドウサイズでフロー制御が行われているにも関わらず、全体としてデータグラム破棄の軽減効果が得られなかったと考えられる。

このことから、提案手法における問題点は、フローごとに独立した輻輳検知を行うために、フロー全体としてルータのバッファ溢れを抑制することができない点であると言える。

この問題を改善するためには、フロー同士の協調のための何らかの仕組みを新たに加える必要があると考えられる。

4 輻輳予測とフロー制御への応用

今回の提案手法において、指摘した問題点を改善し、輻輳通知の時間遅れを短縮することによって、複数フロー下でも輻輳制御に効果的であることが確認できれば、輻輳予測によって更なる輻輳制御の性能改善を見込むことができる。すなわち、輻輳通知の時間遅れが輻輳制御の性能に影響を与えているのであれば、輻輳予測によって時間遅れの影響を可能な限り抑制することができるからである。

ここで重要なのは、輻輳予測手法であるが、これについては今回の提案手法と同様の方法によって、確認応答 (ack) メッセージにパス上のルータの情報 (平均バッファ長など) を書き込むことで、送信側にパス上のルータ情報を集め、その情報に基づき時系列解析等の手法を用いて輻輳を予測することを検討している。

5 おわりに

これまで示したように、明示的輻輳制御で用いられる輻輳通知は本質的に時間遅れを伴うが、その遅れが輻輳制御に大きく影響することが確認された。

そこで、輻輳予測の性能改善を目指して、まず、輻輳通知の時間遅れを短縮する簡易試作モデルを実装

したが、複数フロー下では対象外のフローの影響から期待された効果が得られなかった。

今後は、まず、今回の提案手法の問題点を改善してその効果を確認し、次のステップとして輻輳予測手法をフロー制御に応用した、新たな提案手法の有効性を明らかにしていく予定である。

6 謝辞

本研究は、文部科学省および日本学術振興会の科学研究費補助金 (基盤 B : 課題番号 13450165) の助成により行われた。ここに深謝致します。

参考文献

- [1] 上野英俊, 木村成伴, 海老原義彦: 明示的輻輳通知を用いた TCP の優先輻輳制御方式, 情報処理学会論文誌, Vol.40, No.1, pp.57-65(1999).
- [2] Floyd, S. and Jacobson, V.: Random Early Detection Gateway for Congestion Avoidance, IEEE/ACM Transaction on Networking, Vol.1, No.4, pp.397-413(1993).
- [3] Floyd, S.: TCP and Explicit Congestion Notification, ACM Computer Communication Review, Vol.24, No.5, pp.10-23(1994).
- [4] McCanne, S. and Floyd, S.: Network Simulator ns (version2), <http://www-mash.cs.berkeley.edu/ns/>.
- [5] 三宅優, 神崎昭浩, 加藤聡彦, 鈴木健二: 広域通信における TCP の高速化を目的とした輻輳回避手順の提案, 情報処理学会 マルチメディア通信と分散処理研究会報告集, 88-11, pp.61-66(1998).
- [6] 山本一隆, 砂原秀樹, 尾家祐二: 広域無線通信メディア利用時のゲートウェイ協調による TCP 性能改善, 情報処理学会 マルチメディア通信と分散処理研究会報告集, 91-16, pp.91-96(1999).
- [7] W.R.Stevens: TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley Publishing Company, 1994.