マルチメディア通信と分散処理 86－10
グループウェア 26－10
（1998. 1. 29）

優先度を考慮にいれた輻輳通知方式による フロー制御に関する研究

上野 英俊† 木村 成伴, 海老原 義彦‡

†筑波大学大学院 工学研究科 ‡筑波大学 電子・情報工学系

〒305 茨城県つくば市天王台 1-1-1

{ueno,kimura,ebihara}@netlab.is.tsukuba.ac.jp

あらまし 今日のインターネットの普及により，さまざまなネットワークサービスが提供され，ネットワーク環境はますます多様化している．これに伴い，サービスが要求する通信品質に応じて優先度を設け，これを利用したネットワークのフロー制御を行うことが重要視されている．TCP/IP では，従来のフロー制御方式に加えて，ルータがネットワークの輻輳状況をエンドホストに通知する明示的輻輳通知が提案されているが，本研究ではこの明示的輻輳通知を用いて，各トラヒックの優先度に応じたフロー制御方式を提案する．さらに，提案方式を用いたネットワークモデル上で計算機シミュレーションを行い，従来方式と同等な平均スループットを保ちながら，提案方式による優先制御が可能であることを示す．

キーワード TCP/IP, フロー制御, 明示的輻輳通知, 優先度, IPv6

A Study of Flow Control by Congestion Notification based on Traffic Priority

Hidetoshi Ueno † Shigetomo Kimura, Yoshihiko Ebihara‡

†Graduate School of Engineering, University of Tsukuba

‡Institute of Information Sciences and Electronics, University of Tsukuba

1-1-1 Ten-nodai, Tsukuba, Ibaraki 305, Japan

{ueno,kimura,ebihara}@netlab.is.tsukuba.ac.jp

Abstract As the Internet is much familiar in nowadays, various network services are provided, and the network environment is more and more diversified. Therefore, it is considered very important to control network flow based on the priority according to the communication quality required by each service. For TCP/IP, in addition to the ordinary flow control, the explicit congestion notification, in which the router informs the congestion status to the end hosts, is proposed. This paper proposes a new flow control using this explicit congestion notification based on the priority of each traffic. The computer simulation of the proposed flow control on a network model concludes that the method keeps its average throughput similar to the ordinarily one and is even able to perform the priority control.

key words TCP/IP, Flow Control, Explicit Congestion Notification, Priority, IPv6

1 はじめに

ATM，ギガビットイーサネット，HIPPI，ファイバチャネル等に代表される高速通信回線が，インターネットの普及やマルチメディアアプリケーションの増加とともに，ネットワークインフラとして整備されつつある．また，コンピュータネットワークは，当初予期しない規模に広がりつつあり，今後さらに拡大することが予想される．したがって，通信プロトコルはネットワークの高速化や拡大化，多様化に対処するように変化する必要がある．

ネットワークの拡大化に伴う対処法として標準化団体IETFは，次世代のインターネットプロトコルに適したIPversion6(IPv6)の標準化作業を進めている．これによりIPアドレスは128ビット長に拡大される．IPv6はパケットのルートを決定するルーティングテーブルの肥大化対策や，その他の機能についても拡張や変更を行い，IPversion4(IPv4)での問題点は解決されつつある[1].

また，多様化とは，多種多様なネットワークサービスが提供されることを意味し，これにより新しいアプリケーションの開発や利用が可能になっている．今後この傾向は強まると予測され，トラヒックの爆発的な増加が見込まれる．トラヒックの増加はネットワークの混雑を引き起こし，輻輳を招くため，データの流量制御であるフロー制御を適切に行う必要がある．

ところで，ネットワークアプリケーションとしてはhttp，telnet，ftp，smtp等があるが，これらのアプリケーションは対話性のあるものと無いものに分けられる．これらのトラヒックの性質に着目すると，対話性のあるトラヒックはリアルタイム性が求められ，対話性のないものと比べると優先度が高いと考えられている．本稿ではルータによるエンドノードへの明示的輻輳通知(ECN:Explicit Congestion Notification)を利用し，高優先のトラヒックは輻輳発生時にも高スループットを維持することが可能なフロー制御方式を提案し，計算機シミュレーションによって評価を行う．

2 TCP/IPのフロー制御方式

インターネットで用いられているTCP/IPでは，TCP層においてフロー制御を行い，他に再送制御や順序制御，エラー検出等を行う信頼性のあるストリーム転送を支援する．

TCPのフロー制御では，タイムアウトによる輻輳検知と，可変長のバイトサイズ指定のスライディングウインドウ制御により，トラヒックシェーピングを行っている．以下でTCPのウインドウ制御の概要を説明する．送信開始直後はスロースタートモードとなり，最初に輻輳ウインドウ(cwnd)は最大セグメントサイズ(MSS)にセットされ，確認応答(ACK)受信により送信終了を確認したセグメントサイズ分づつcwndを増加する．これによりcwndは指数的に増加する．cwndがスロースタートスレッショルドを越えると輻輳回避モードに入り，ACK受信によりウインドウサイズを少しずつ増加させる．送信側はTCPヘッダにより受信側から通知されるウインドウサイズの値とcwndの内，小さい値を用いる．また，一定時間内にACKが戻らないと送信タイムアウトとなり，cwndをMSSに戻し，スロースタート動作を再開する．以上の制御によりネットワークが非輻輳時にはウインドウサイズを大きく取り，逆に輻輳時にはウインドウサイズを下げる動的な制御を行う[2].

2.1 現状のフロー制御の問題点とその解決法

送信側は送信タイムアウトによりセグメント[1]の再送を行い，セグメントの廃棄やネットワークの輻輳が発生したことを判断する．これは内部のネットワークの輻輳状況を予想した制御方法であり，**暗示的輻輳通知**と呼ぶ．暗示的輻輳通知は輻輳発生時の性能回復が迅速でなく，さらに通知内容が実際のネットワークの状況を正確に反映していないため，スループットの低下を招くことが問題である．伝播遅延時間が大きい広域網ではこの問題が特に顕著に現れる．輻輳検知の精度を向上するための再送制御機構として，早期再送(Fast Retransmit)や早期回復(Fast Recovery)[3]，選択的確認応答(SACK:Selective Acknowledgement)[4]機能があるが，これらの機能を用いたとしても，暗示的輻輳通知での問題点を完全に解決するには至らない．

これに対しルータが輻輳状態であることをエンドホストへ通知する制御を加え，通知された輻輳情報を利用して輻輳回避を行う方式が提案されている．これを**明示的輻輳通知**と呼ぶ．ただし，ルータはTCP/IPプロトコルのIP層までの制御のみを行うため，輻輳情報はデータグラムと共に送信され，エンドホストでTCP層に渡される．

データリンク層レベルで明示的輻輳通知を用いるプロトコルとして，フレームリレーやATMのFECN(Forward ECN)，BECN(Backward ECN)があるが，これらの輻輳制御では明示的輻輳通知による有意な性能向上が見られる．したがって，データリンク層より上位のTCPコネクションで明示的輻輳通知を行うことは，輻輳制御の性能向上につながると期待される．

[1]TCPのデータパケットをセグメント，IPのデータパケットをデータグラムと使い分ける．

3 明示的輻輳通知

TCP/IP の明示的輻輳通知には，ERD(Early Random Drop Gateway)，ICMP 始点抑制，DECbit アルゴリズム [5] 等が提案されている．また，インターネット標準になりつつある RED ゲートウェイ (Rondom Early Detection Gateway) 方式 [6] は，論文 [5] の方式を変更したものである．各明示的輻輳通知方式は，送信側がネットワークの輻輳をいち早く検知することができるため，スループットが向上する等の改善がされている [7].

RED ゲートウェイ方式

RED ゲートウェイ方式について説明する (図 1)．データグラムヘッダに輻輳通知のための輻輳通知ビット (ECN ビット) フィールドを設け，送信ノードは 0 に設定してデータグラムを送信する．後述の手順によりルータが輻輳と判断すると ECN ビットを 1 にセットする．受信側は ECN ビットがセットされたデータグラムを受け取ると，それに対する ACK 中の ECN ビット[2]をセットし，送信側へ報告する．送信側の TCP で CI ビットを利用するには以下に従う．(a)CI ビット受信に対する TCP の処理はタイムアウトに対する処理と同様にする．(b)TCP は 1 ラウンドトリップタイム当たり，高々一回だけ CI ビットに応答する．TCP 送信側は直前のラウンドトリップタイムに CI ビットやタイムアウトに応答していたならば，そのラウンドトリップタイム間に到着した CI ビットは無視する．(c)TCP は受信 ACK に対するデータパケットの送信に関して既存のアルゴリズムに従う．タイムアウトに対する処理も既存のアルゴリズムに従う．

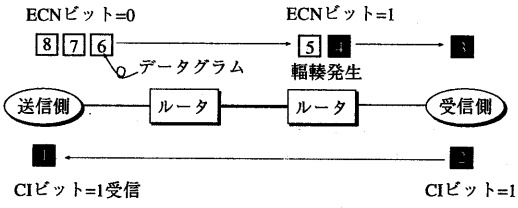

図 1: RED ゲートウェイ方式の輻輳通知

次にルータにおいて ECN ビットをセットする手順を述べる．ルータでは過去の履歴と現在のバッファ長に基づき，平均バッファ長 avg を計算する．avg は新しいデータグラム到着毎に更新される．また，バッファに関して max_{th} と min_{th} の 2 つの値を持つ．$avg < min_{th}$ の時には ECN ビットはセットせず，$max_{th} \leq avg$ の時には ECN ビットをセットする．$min_{th} \leq avg < max_{th}$ の時は，確率 p で ECN ビットをセットする．データグラムがバッファを占める割合が高い場合や ECN ビットをセットしない期間が長く続いた時には p の確率は高くなるよう計算される [6].

RED ゲートウェイ方式では，適切な確率 p でデータグラムに ECN ビットをセットすることで，長期間にわたっての平均キューサイズが制御可能となる．また，min_{th} と max_{th} の値により伝播遅延の調節も可能になる．さらに，あるコネクションのデータグラムがバースト的にルータに到着した場合でも，特定のコネクションからのデータグラムが連続してバッファ溢れを起こし，バーストデータの大部分が廃棄されることが少なくなるという効果もある．

4 優先度を考慮にいれた輻輳通知

ECN ビットと CI ビットを用いて輻輳通知を行う RED ゲートウェイ方式を用い，優先度に応じてスループットを変化させるフロー制御方式を提案する．以下に提案方式を述べる．

RED ゲートウェイ方式において平均バッファ長 avg が $min_{th} \leq avg < max_{th}$ の関係にある時は ECN ビットをセットする確率 p を変化させる．提案方式では，高優先のデータグラムには ECN ビットのセット確率を下げ，逆に低優先のものについては相対的に ECN ビットセットの確率を高める．これにより高優先のデータグラムの送信側は cwnd を下げる確率の減少につながり，低優先のものについてはこの逆になるため，優先度に応じたスループットの変化が得られる．

優先度や輻輳状況の通知にはデータグラムヘッダを用いる．その際，本稿では，標準化されつつある IPv6 を対象とすることにする．優先度の通知には IPv6 ヘッダの優先度フィールドを用い，ECN ビットや CI ビットは，IPv6 ヘッダのオプションとして輻輳制御ヘッダを新たに提案することで実現する．

優先度フィールド

IPv6 ヘッダを図 2に示す．優先度フィールドは 4 ビットから構成され，同じ発信者からの他のデータグラムと比較して，その望ましい配送優先権を識別することができるようにする．値 8 ～ 15 が相対的な優先度を表し，値 0 ～ 7 は別の目的で使用される．なお，値 15 の優先度が一番高く，値 8 の優先度が一番低い．

[2]本稿では，ACK 中の ECN ビットを分かりやすく区別するために輻輳表示 (CI:Congestion Indication) ビットとする．

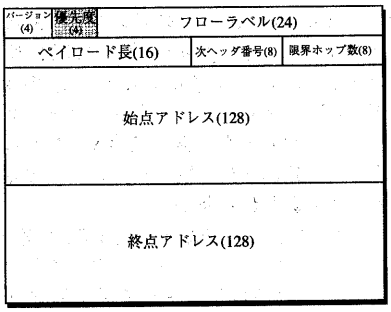

図 2: IPv6 の標準ヘッダ

輻輳制御ヘッダオプション

提案する IPv6 の拡張ヘッダである，輻輳制御ヘッダのフォーマットを図 3に示す．ヘッダは 8 バイトで構成され，将来の拡張のための予約フィールドや優先度の重み，各種フラグから成る．

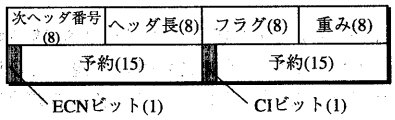

図 3: 提案する拡張輻輳通知ヘッダ

ECN ビットのセット確率 p

ECN ビットをセットする確率 p には以下の式を用いる．

$$N = p \times \frac{(15 - pri)^w}{(15 - b)^w}$$

N ：新しい廃棄確率 ($N_{max} = 1$)

p ：RED ゲートウェイアルゴリズムで与えられる廃棄確率 p

pri ：データグラムの優先度

b ：基準となる優先度

w ：優先度の重み

基準となる優先度の値を境に優先度の大小が区別される．ここでは IPv6 ヘッダの 8 ～ 15 までの 8 段階の数値を扱い，優先度 15 は特に重要度が高いデータグラムに対して付するので b としては 11 を用いる．優先度の重みについては値が大きいほど確率 p の変動率が上がる．なお，優先度フィールドを使用しない場合には $w = 0$ にすることにより $N = p$ が得られる．

5 シミュレーションによる評価

以上の提案をネットワークシミュレータ [8] を用いた計算機シミュレーションにより実験を行う．ネットワークモデル図 4に示す．表 1はシミュレーションモデル上で扱う ftp コネクションの優先度を表す．

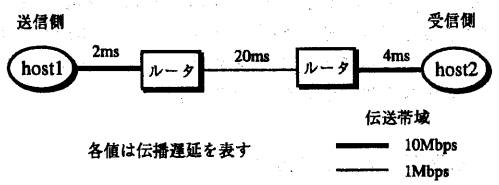

図 4: ネットワークモデル

コネクション	名前	アプリケーション	優先度
host1→host2	ftp1	優先度の低い ftp	9
host1→host2	ftp2	通常の ftp	11
host1→host2	ftp3	通常の ftp	11
host1→host2	ftp4	優先度の高い ftp	13

表 1: コネクションと優先度

優先度の低い ftp の例としては，時間的制約が無いデータを anonymous ftp で転送することや，smtp や nntp による転送が考えられる．優先度の高い ftp としては，時間的制約があるデータを転送する場合や，telnet，http など対話性のあるトラヒックの転送を当てはめることができる．

各実験に共通なシミュレーション条件は次のとおりである．

- TCP には 4.3BSD Reno を用いる．
- ACK のサイズは 68byte とする．これは標準 TCP ヘッダ (20byte) と標準 IPv6 ヘッダ (40byte) に加えた輻輳制御ヘッダ (8byte) を加えた総和である．
- データグラムサイズは 512byte とする．インターネットのトラヒックのデータグラムサイズは大部分が 512byte であることを用いた [9]．
- 定数 $w = 1$, $min_{th} = 12 \times 512\text{byte}$, $max_{th} = 3 \times min_{th}$

5.1 受信セグメント数の比較

図 5にコネクション開始から 60 秒間の受信セグメント数を示す．ここでは ftp1(優先度 9) と ftp4(優先度 13) の二つのコネクションのみを示した．

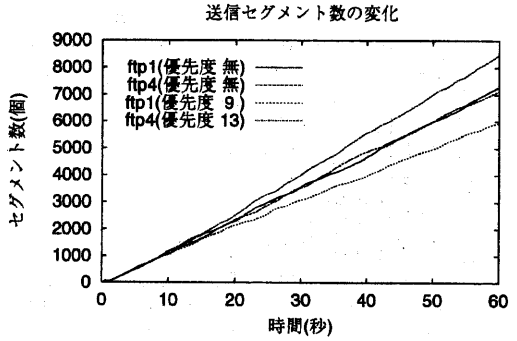

図 5: 受信セグメント数の比較

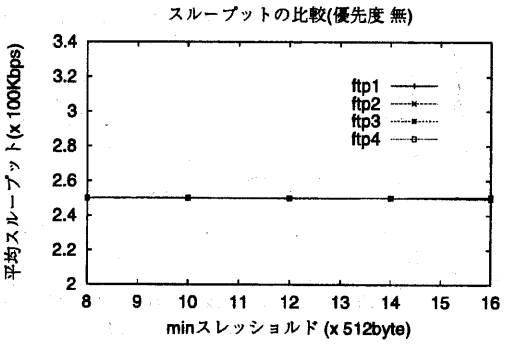

図 6: min_{th} とスループットの比較 (優先度 無)

優先度により受信セグメント数に差が見られるため，優先度を考慮したフロー制御が行われていることが確認できる．

5.2 min_{th} の変化による比較

ルータのバッファにおける min_{th} を 4096byte, 5120byte, 6144byte, 7168byte, 8192byte と変化させた場合のスループットを比較する．図 6は優先度を考慮せず，図 7は優先度を考慮している．

優先度を考慮しない場合の各コネクションにスループットの差は無いが，優先度を考慮する場合は優先度に応じたスループットの変化が確認できる．ただし，両者の平均スループットはほぼ同じ値となった．また，min_{th} を大きくするにしたがいECN ビットをセットする確率が下がるため，高優先と低優先のデータグラム間のスループットの差が小さくなる．これは，非輻輳時に優先度の低いデータグラムの送信を制限しないことを表す．

5.3 データグラムサイズの変化による比較

送信データグラムサイズを 512byte, 1024byte, 1536byte, 2048byte と変化させた際のスループットの変化を比較する．図 8は優先度を考慮せず，図 9は優先度を考慮している．

優先度を考慮しない場合には各コネクションにスループットの違いは見られないが，優先度を考慮する場合はスループットの変化が確認できる．なお，データグラムサイズが大きくなるほど，再送オーバヘッドが大きいため，一般的にスループットは低下する．論文 [9] より TCP/IP の多くのデータグラムサイズは 512byte であるため，ユーザはデータグラムサイズによる再送オーバヘッドの相違を意識せずに優先度を割り当て，それに伴ったスループットを得ることができる．

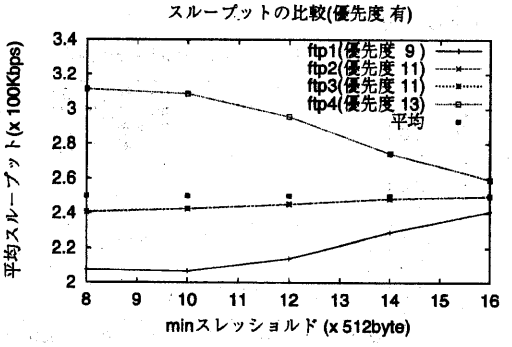

図 7: min_{th} とスループットの比較 (優先度 有)

5.4 重みの変化による比較

図 10は優先度の重み w を変化させた時の各優先度におけるスループットを示している．

優先度の重み w を高くすると，高優先と低優先データグラムのスループットの差が大きくなる．ユーザは特に高スループットを得る必要のあるセグメントを送信したい場合は，w の値を高くする．また，優先度を考慮することによる，全体の平均スループットは優先度を考慮しない場合とほぼ同じ結果が得られた．

6 まとめと今後の課題

本稿では，明示的輻輳通知を用いて，優先度に応じたフロー制御を行う方式を提案し，計算機シミュレーションによりその評価を行った．提案方式により高優先と低優先のデータグラムにスループットの差が生じることを確認した．その際，優先度を考慮した場合と考慮しない場合について，全体の平均スループットはほぼ等しいという結果が得られた．

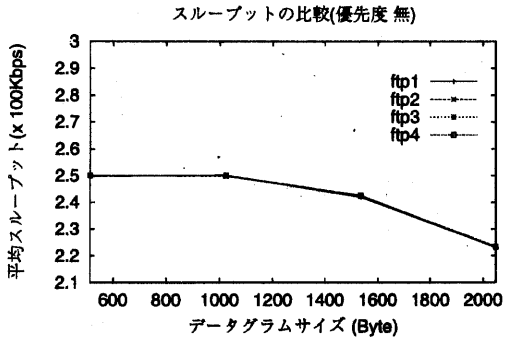

図 8: データグラムサイズとスループットの比較 (優先度 無)

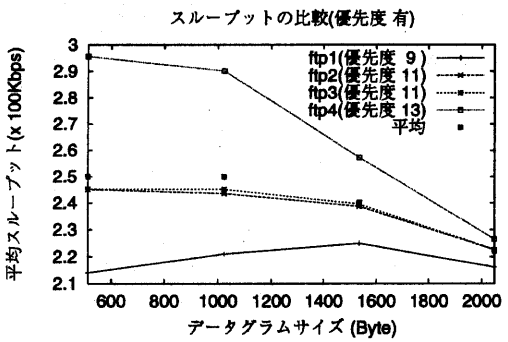

図 9: データグラムサイズとスループットの比較 (優先度 有)

ユーザは得られるスループットの特性を理解した上で，優先度を与える必要があり，優先度の値の選択は難しい．また，得られたスループットの妥当性の判断はユーザにまかせられる．一般的なユーザは高優先のデータの高いスループットを期待しており，提案方式はその要求を満たすといえる．

ユーザーが与えた優先度に偏りがあり，低優先のデータのスループットが極端に下がるような場合には，一時的に低優先のデータのスループットを上げるための制御を加えることや，偏りが無くなるようユーザに報告するといった考慮が必要である．ユーザは自分のデータに対しては高いスループットを得たいという希望があるため，このような必要性が頻繁に発生する可能性がある．この制御方法とその評価については今後の課題である．また，今回用いたRED ゲートウェイ方式以外の明示的輻輳通知方式についても，提案方式と同様に優先度を考慮することで，同様な結果が得られるものと期待できるので，これについても検討したい．

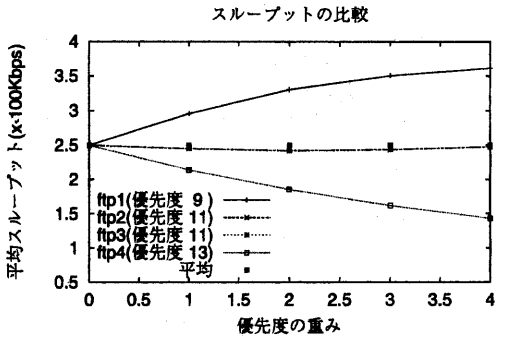

図 10: 優先度の重みによるスループットの比較

参考文献

[1] "Internet Protocol, Version 6 (IPv6) Specification". *RFC-1883*, 1995.

[2] Van Jacobson. "Congestion Avoidance and Control". *Proc. of ACM SIGCOMM'88*, pages 314–329, 1988.

[3] "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms". *RFC-2001*, 1997.

[4] "TCP Selective Acknowledgement Options". *RFC-2018*, 1996.

[5] K. K. Ramakrishnan and Raj jain. "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer". *Proc. of ACM SIGCOMM'88*, pages 303–313, 1988.

[6] Sally Floyd and Van Jacobson. "Random Early Detection Gateways for Congestion Avoidance". *IEEE/ACM Transactions on Networking*, Vol.1(No.4):397–413, August 1993.

[7] Sally Floyd. "TCP and Explicit Congestion Notification". *ACM Computer Communication Review*, Vol.24(No.5):10–23, 1994.

[8] Network Simulator -ns(version2). `http://www-mash.cs.berkeley.edu/ns/`.

[9] 串田高幸. "インターネットのTCP トラヒックの解析". 情報処理学会研究報告 マルチメディアと分散処理, Vol.84(No.4):19–24, 1997.