

明示的輻輳通知を用いた TCP の優先輻輳制御方式

上野 英俊[†] 木村 成伴^{††} 海老原 義彦^{††}

インターネットで用いられている TCP では、タイムアウトによる輻輳検知を用いてエンドトゥエンドでの輻輳制御を行う。このため、特に伝播遅延の大きい広域網においては、ネットワーク内部の輻輳状況が輻輳制御に正確に反映されないという問題点がある。これを改善するため、従来方式に加え、ルータがネットワーク内の輻輳状況をエンドホストに通知し、これに応じてウインドウサイズを調節する明示的輻輳通知方式が提案されており、その代表的な方式の1つとして RED ゲートウェイがある。ところで、ネットワークサービスで扱うデータは主にリアルタイム性を要求する音声や動画などと、そうでないデータ通信などに分類される。これを利用し、サービスが要求する通信品質に応じて優先順位付けを行い、これに応じた輻輳制御を行えば全体的な利用効率の向上が期待できる。そこで本論文では、まず、優先順位として IPv6 の優先度を対象とし、優先制御のための輻輳通知用拡張ヘッダを定義する。次に、これを用いた RED ゲートウェイによる明示的輻輳通知方式を改良し、優先順位に応じた輻輳制御を行う優先輻輳制御アルゴリズムを提案する。最後に、実環境に近い条件の下で、計算機により本アルゴリズムのネットワークシミュレーションを行う。この結果から、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し、本方式の優位性を立証する。

Prioritized Congestion Control with Explicit Congestion Notification for TCP

HIDETOSHI UENO,[†] SHIGETOMO KIMURA^{††} and YOSHIHIKO EBIHARA^{††}

In the TCP, the congestion detection by communication timeout is used for the end-to-end congestion control. However, it is not able to reflect the exact congestion status in the inside of the network, especially to wide area networks. To improve this problem, in addition to the existent methods, the explicit congestion notification is proposed to notify the congestion status through routers in the network. The one of its famous methods is the RED (Random Early Detection) gateway. By the way, in general, communicated data are classified into delay sensitive ones and others. If the congestion control reflects the priorities with respect to such communication qualities required by services, it can be expected to improve the utilization of the entire network. This paper expands the explicit congestion notification of the RED gateway for prioritized communications. The priorities are expressed by the one of the IPv6. The congestion control extension header is defined and the congestion control algorithm for the priorities is proposed. Finally, the network simulation under the nearer conditions to the real network concludes that the proposed method can provide the priority control on data traffic, while it keeps the average throughput similar to the one of the ordinary ones.

1. はじめに

近年、インターネットの急激な普及にともない、多種多様なネットワークサービスが提供され、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。輻輳はネットワークのスループットを減少させるため、輻輳制御を適切に行う必要がある。

インターネットで用いられている TCP では、タイムアウトによる輻輳検知とスロースタートを用いたウインドウ制御によるトラヒックシェーピングにより、これを実現している。従来、この制御はエンドトゥエンドで行われているが、特に広域網においては、ネットワーク内部の輻輳状況が輻輳制御に正確に反映されないという問題点がある。そこで、従来方式に加え、

[†] 筑波大学大学院工学研究科

Graduate School of Engineering, University of Tsukuba

^{††} 筑波大学電子・情報工学系

Institute of Information Sciences and Electronics, University of Tsukuba

ルータがネットワーク内の輻輳状況をエンドホストに通知し、ウィンドウサイズを調節する明示的輻輳通知 (ECN: Explicit Congestion Notification) が提案されている。

ところで、ネットワークアプリケーションとして代表的なものに、http (Hypertext Transfer Protocol)¹⁾, telnet²⁾, ftp (File Transfer Protocol)³⁾, smtp (Simple Mail Transfer Protocol)⁴⁾, nntp (Network News Transfer Protocol)⁵⁾ などがあるが、これらのアプリケーションは対話性のあるものかないものに分けられる。これらのトラフィックの性質に着目すると、対話性のあるトラフィ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。また、リアルタイム性が要求される動画や音声についても優先度が高いと考えられている。これを利用し、各サービスが要求する通信品質に応じて優先度を設け、これに応じた輻輳制御を行えば全体的な利用効率の向上が期待できる。

本論文ではルータによるエンドノードへの明示的輻輳通知を利用し、各トラフィックの優先度に応じた輻輳制御方式を提案する。さらに、従来方式と同等な平均スループットを保ちながら、高優先のトラフィックが輻輳発生時にも高スループットを維持することが可能であることを、計算機シミュレーションによって示す。

以下では、まず、2章でTCPの輻輳制御について説明し、既存システムの問題点を示した後、明示的輻輳通知について説明する。3章で明示的輻輳通知方式を用いて優先制御を行う輻輳制御方式を提案する。4章では、計算機シミュレーションによる評価と考察を行う。最後に5章で今後の検討課題と結論について述べる。

2. TCP/IPの輻輳制御

2.1 TCPにおける暗示的輻輳通知

TCPでは信頼性のあるストリーム転送をエンドトウエンドで提供するため、輻輳制御を行い、ほかに再送制御や順序制御、エラー検出などを行う⁶⁾。TCPのウィンドウサイズは可変長であるため、ネットワークが非輻輳時にはウィンドウサイズを大きくとり、逆に輻輳時にはウィンドウサイズを小さくするといった動的な制御を行っている。このTCPの輻輳制御についての詳細は文献7)を参照されたい。

TCPでは、送信側はタイムアウトにより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し、セグメントの再送を行う。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

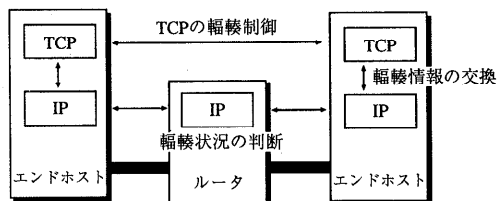


図1 明示的輻輳通知

Fig. 1 Explicit congestion notification.

輻輳通知と呼ばれる。この通知方式で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。また、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。インターネットの急速な拡大により伝播遅延時間はますます増大することが予想され、この問題を改善することは急務となっている。これをふまえ、ネットワーク内の輻輳状況を陽に取得し、この情報に基づく制御方式が注目されている。次節でこの制御方式について述べる。

2.2 TCP/IPにおける明示的輻輳通知

前節で述べた暗示的輻輳通知に対し、明示的輻輳通知は、ルータで軽度の輻輳が発生している、もしくは、輻輳発生の前兆があることを、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早期に輻輳回避を行う方式である。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は、フレームリレーやATMのFECN (Forward ECN), BECN (Backward ECN) などがあるが、これらの輻輳制御では明示的輻輳通知により、有意な性能向上が見られる⁸⁾。TCP/IPではルータは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ともに送信され、エンドホストでTCP層に渡される。図1はルータにより検出された輻輳情報をIPデータグラムによりエンドホストまで運搬し、輻輳制御を行う上位のTCPまで輻輳情報を受け渡している様子を簡単に表している。

TCP/IPの明示的輻輳通知には、ERD (Early Random Drop Gateway)⁹⁾, ICMP (Internet Control Message Protocol) 始点抑制¹⁰⁾, DECbit アルゴリズム¹¹⁾などが提案されている。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などの改善がなされている¹²⁾。以下では各明示的輻輳通知方式について簡単に説明する。

ERDは、ルータのバッファサイズが廃棄レベルに達したときに、バッファ中にあるランダムに選択され

たデータグラムを廃棄する。これによりタイムアウトが発生し、ウィンドウサイズを下げるため、輻轉回避につながる。ERDでは、各コネクションへのトラフィックの公平な割当てが可能であるが、輻轉通知がタイムアウトを用いるので、再送が必要である。また、複数のコネクションが同時にタイムアウトを発生させた場合には、必要以上にスループットを下げるという欠点がある⁹⁾。

ICMP 始点抑制メッセージは、ルータにおいてデータグラムの廃棄が発生すると、ルータは ICMP 始点抑制メッセージを発行し、これを送信側に通知する。ICMP 始点抑制メッセージを受信した IP 送信側は、上位層にその情報を渡すが、その用法については上位層で定義する¹⁰⁾。この方式では輻轉時に ICMP メッセージを生成することから、輻轉状況の悪化を助長するという問題点がある。また、データグラムの廃棄が発生した後に ICMP メッセージを生成することから、その情報が送信側に報告されたときにはネットワークの輻轉状況がすでに改善されているといった場合が考えられる。

DECbit アルゴリズムは、輻轉表示ビットをパケットヘッダに設け、ネットワークの輻轉状況をエンドホストに通知する方式である。ルータは、パケット到着時に計算した平均バッファ長が閾値を超えたときに、中継するパケットに輻轉表示ビットをセットする。送信側では、2 ラウンドトリップごとに輻轉表示ビットの情報に基づいてウィンドウサイズを次のように更新する。すなわち、直前のウィンドウにおいてパケットの少なくとも半分に輻轉表示が設定されていたら、ウィンドウサイズを指数的に減少し、そうでなければ線形的に増加する¹¹⁾。DECbit アルゴリズムは、パースト的なトラフィックに対して輻轉表示ビットのセットに偏りが生じることが文献 13) で指摘されている。

これに対し、データグラム廃棄が発生する前に輻轉情報を報告する方式として、インターネット標準になりつつある RED ゲートウェイ (Random Early Detection Gateway) 方式¹³⁾がある。これは文献 11) の方式を変更したものである。2つの方式の詳細な相違については文献 13) で述べられている。

RED ゲートウェイ方式を用いて輻轉通知を行う手順は、文献 12) で提案されている。図 2 を用い、その概要を述べる。まず、データグラムヘッダに輻轉通知のための輻轉通知ビット (ECN ビット) フィールドを設け、送信側ではこれを 0 に設定してデータグラムを送信する (図 2 データグラム 8~6)。後述の RED ゲートウェイアルゴリズムによりルータが輻轉発生

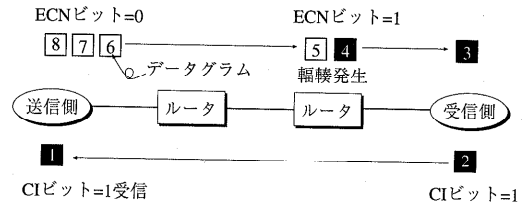


図 2 RED ゲートウェイ方式の輻轉通知

Fig. 2 Congestion notification in RED gateway method.

前兆であると判断すると ECN ビットを 1 にセットする (図 2 データグラム 4)。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 (図 2 データグラム 3)、それに対して確認応答 (ACK) 中の ECN ビットをセットし (図 2 データグラム 2)、送信側へ報告する。送信側で CI ビットを受信すると (図 2 データグラム 1)、その旨を TCP へと報告し、文献 12) に従う手順によりウィンドウサイズを減少させ、ネットワークに送信するセグメント数の制限を行う。

ECN ビットは、ルータにおいて、文献 13) で提案されている RED ゲートウェイアルゴリズムに従ってセットされる。ルータでは過去の履歴と現在のバッファ長に基づき、データグラム到着ごとに平均バッファ長 avg を計算する。バッファ長に関してあらかじめ定義した max_{th} と min_{th} と比較し、 $max_{th} \leq avg$ のときには ECN ビットをセットし、 $avg < min_{th}$ のときには ECN ビットはセットしない。 $min_{th} \leq avg < max_{th}$ のときは、確率 p_a で ECN ビットをセットする。ただし、 p_a は式 (1)、(2) で計算される。ここで、 max_p は p_b の上限値、 $count$ は最後に ECN ビットがセットされてから送出した ECN ビットがセットされないデータグラム数のカウントである。データグラムがバッファを占める割合が高い場合には式 (1) により、また、ECN ビットをセットしない期間が長く続いたときには式 (2) によって p_a の確率は高くなるよう計算される。

$$p_b = max_p \frac{avg - min_{th}}{max_{th} - min_{th}} \quad (1)$$

$$p_a = \frac{p_b}{1 - count \cdot p_b} \quad (2)$$

RED ゲートウェイ方式では、適切な min_{th} と max_{th} の値により長期間にわたっての平均キューサイズが制御可能であり、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パースト

* 以下では、データグラムの ECN と区別するために、以下では輻轉表示 (CI: Congestion Indication) ビットと呼ぶ。

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¹⁴⁾。

ところで、1章で述べたように、各トラフィックはその性質により優先度を設けることができる。この優先度を考慮するよう確率 p_a を定義すれば、全体的な通信品質を向上させることが期待できる。次章では、RED ゲートウェイアルゴリズムにおける新たな p_a による優先輻輳制御方式を提案する。

3. 優先度を考慮にいたれた輻輳通知

RED ゲートウェイアルゴリズムにより優先度に応じた輻輳制御を行うため、高優先のデータグラムには ECN ビットのセット確率 p_a を下げ、逆に低優先のものについては相対的に確率 p_a を高める方式を提案する。具体的な p_a の決定方法については 3.3 節で後述する。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、本論文では、標準化された IP version6 (IPv6)¹⁵⁾ を対象とすることにす。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設定することで実現する。

3.1 優先度フィールド

IPv6 ヘッダの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。優先順位の値は 0~15 までであり、このうち 8~15 までの値は相対的な優先度を表す。値 15 の優先度が一番高く、値 8 の優先度が一番低い。0~7 までの値は別の目的で使用され、特定のアプリケーションに対して特別に割り当てられる¹⁵⁾。このため、以下では一般的に利用できる 8~15 までの値のみを優先度として使用することとした。この 8 段階の優先度では不十分な可能性も考えられるが、これは IPv6 での制約であり、本論文での考察の対象とはしない。

3.2 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。本拡張ヘッダは 8 オクテットで構成される。次ヘッダ番号は、輻輳制御ヘッダのすぐ後に続いているヘッダのタイプを識別する。また、ヘッダ長は (ヘッダの総バイト数 - 8) となるため、提案ヘッダでは 0 になる。以上の 2 つのフィールドの使用法は、ルーティングヘッダなどの他の拡張ヘッダと同様である。次に、輻輳状態の通知のため、

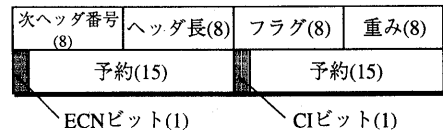


図 3 提案する拡張輻輳通知ヘッダ

Fig. 3 Proposed extension congestion notification header.

ECN ビットと CI ビットを 1 ビットずつ用意する。さらに、重みフィールドを 8 ビット用意し、最小 0~最大 255 までの値を送信側のユーザがセットする。これは、優先度の重みを考慮に入れるために用いられる。また、将来の拡張のための予約フィールドや各種フラグを設ける。この 2 つについては今回は使用法を定義しない。

3.3 ECN ビットのセット確率

RED ゲートウェイアルゴリズムにおいて、 $min_{th} \leq avg < max_{th}$ のときに、ECN ビットをセットする確率の算出には式 (3) を用いる。

$$p_{new} = p_a \frac{(15 - pri)^w}{(15 - b)^w} \quad (3)$$

- p_{new} : 提案式により新たに得る ECN ビットのセット率 $MAX(p_{new}) = 1$
- p_a : 式 (1), 式 (2) の RED ゲートウェイアルゴリズムで与えられた ECN ビットのセット率
- pri : IPv6 の優先度フィールドで与えられるデータグラムの優先度
- b : 基準となる優先度
- w : 提案した輻輳制御ヘッダの重みフィールドで与えられる優先度の重み

基準となる優先度 b の値を境に優先度の大小が区別される。ここでは優先度として 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ことにするので、本論文では b として 8 と 14 の中間の値である 11 を用いる。この理由は、本論文では、各データグラムに与えられた優先度の分布が一様である状況を想定しているためである。この場合、優先度の平均値 (=11) 以外の値を基準として用いると、優先度が 8 段階しかないためデータグラムの廃棄率に大きな偏りが生じ、ユーザが求めた優先度と実際に得たスループットに隔たりが生じる場合があるため適切でない。

優先度の重み w は値が大きいほど確率 p_{new} の変動率が上がる。ただし、 $w = 0$ ならば、 $p_{new} = p_a$ となるため、優先制御を行わない。表 1 は、 w を変化させたときの p_{new} の p_a に対する比率を優先度 9, 11,

表1 p_{new} の p_a に対する比率
Table 1 The ratio of p_{new} to p_a .

重み	優先度 9	優先度 11	優先度 13
$w = 0$	1.00	1.00	1.00
$w = 1$	1.50	1.00	0.50
$w = 2$	2.25	1.00	0.25
$w = 3$	3.375	1.00	0.125
$w = 4$	5.0625	1.00	0.0625

13 の場合について比較している。表 1 より、 w が大きくなるに従い、優先度 9 の p_{new} は上昇することが分かり、そのスループットは低下することが予想される。同様に、優先度 13 の p_{new} は下降するので、そのスループットは上昇すると予想できる。また、 $w = 0$ と $w = 1$ 以外の値の場合には、ECN ビットのセット確率の平均値が上昇する。たとえば、式 (3) において $w = 4$ とするとき、 p_a に対する優先度 9、13 のときの p_{new} の比率は、それぞれ 5.0625、0.0625 であり、優先度 9 の上昇率が優先度 13 の下降率よりも大きい。この場合、低優先のデータグラムに高確率で ECN ビットがセットされることになるが、ECN ビットがセットされる頻度が上がると、ウィンドウサイズが小さくなる頻度も高くなる。このため、ルータのバッファを占めるデータグラム量が減少し、式 (1) により p_b が減少する。さらに、式 (2) により、ECN ビットセット直後には $count$ の値は 0 にリセットされ p_a も減少することになる。 p_a の減少により、次に ECN ビットをセットするまでの期間の伸びが期待できることから、結果として ECN ビットセット率はそれほど上昇せずに調整されると考えられる。

4. シミュレーションによる評価

前章で提案した優先度を考慮にいたれた輻輳通知方式の有効性を確認するため、ネットワークシミュレータ ns¹⁶⁾を用い、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を図 4 に示す。送信側はつねに送信データを保持し、ウィンドウサイズの範囲内でセグメントを送出し続ける状況を作る。これはボトルネックとなる 1Mbps のリンクが、つねに送信データを保持している状態にするためである。送信側が保持するデータを連続的に送信するために、ftp によるデータの転送を行う。実験では、ネットワークモデル上に 4 つの ftp コネクションを用意し、それぞれ表 2 に示す優先度を持つ。なお、表 2 での ftp1 などの表記は ftp コネクションの識別名を表す。

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。

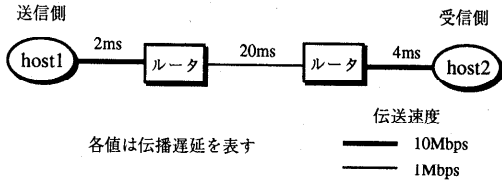


図 4 ネットワークモデル
Fig. 4 Network model.

表 2 コネクションと優先度
Table 2 Connection and priority.

コネクション	識別名	ftp の種類	優先度
host1 → host2	ftp1	低優先の ftp	9
host1 → host2	ftp2	通常の ftp	11
host1 → host2	ftp3	通常の ftp	11
host1 → host2	ftp4	高優先の ftp	13

- 4.3BSD Reno で実装された TCP を用いる。
- ACK のサイズは 68 オクテットとする。これは標準 TCP ヘッダ (20 オクテット) と標準 IPv6 ヘッダ (40 オクテット) に輻輳制御ヘッダ (8 オクテット) を加えた総和である。
- $max_p = 1/50$ とする。これは文献 13) の実験で用いられた値である。
- 精度検定は信頼区間 95% のバッチ平均法により行い、精度は 5% 以内とした。

以下の実験では次に示す条件を基に、 min_{th} 、データグラムサイズ、優先度の重み w のそれぞれを変化させたときのスループットの様子について評価する。

- データグラムサイズは 512 オクテットとする。インターネットのトラフィックのデータグラムサイズは大部分が 512 オクテットであることを用いた¹⁷⁾。
- $w = 1$ とする。重みの影響が最も小さい $w = 1$ で他のパラメータに関する実験を行い、最後の実験で w を変化させたときの比較を行う。
- $min_{th} = 12 \times 512$ オクテットとした。また、文献 13) では、 max_{th} は最低でも min_{th} の 2 倍が有益だとしてあり、文献 13) の実験でも用いられた $max_{th} = 3 \times min_{th}$ とする。

4.1 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 2 つのコネクションに対して、それぞれ優先制御前後の 2 種類の方式の合計 4 つのグラフを示す。

図 5 において、ftp1 では、開始から 30 秒後、および、60 秒後に受信したセグメント数は、優先制御がなしの場合にそれぞれ 3565 個 (1825 Kbyte)、および 7280

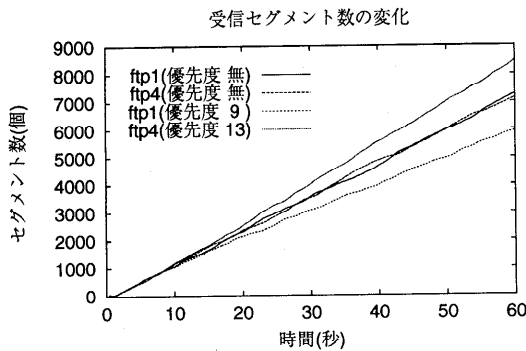


図5 受信セグメント数の比較
Fig. 5 Comparison of received segment number.

個 (3727 Kbyte) であるのに対し、優先度9では3081個 (1577 Kbyte)、および5933個 (3038 Kbyte)となり、優先制御がなしの場合の86.4%、および81.5%に低下している。また、ftp4は、優先制御がなしの場合の3532個 (1808 Kbyte)、および7141個 (3656 Kbyte)から、優先度13では4016個 (2056 Kbyte)、および8488個 (4346 Kbyte)となり、優先制御がなしの場合の113.7%、および118.9%に上昇している。以上から、低優先分のスループットが高優先のそれに割り当てられていることが分かり、優先度を考慮した輻輳制御が行われていることが確認できる。

4.2 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4096 オクテット、5120 オクテット、6144 オクテット、7168 オクテット、8192 オクテットと変化させた場合のスループットを比較する。図6に優先度を考慮しない場合を、図7に優先度を考慮した場合のシミュレーション結果を示す。

図6から、優先制御前には4つのコネクションでそれぞれ250 Kbpsのスループットが得られており、すべての min_{th} について等しい。これは、ボトルネックとなっている1 Mbpsの帯域を4つのコネクションで均等に利用しているからである。

図7より、 min_{th} を大きくするに従い、高優先のスループットは減少するが、低優先のそれは増加し、高優先と低優先のスループットの差が小さくなる。たとえば、 $min_{th} = 4096$ オクテットのとき、ftp1とftp4のスループットはそれぞれ208 Kbps、311 Kbpsであり、図6の結果(250 Kbps)との比率はそれぞれ83.2%、124.4%となるのに対し、 $min_{th} = 8192$ オクテットでは241 Kbps、259 Kbpsにとどまり、比率もそれぞれ96.4%、103.6%である。これは、 min_{th} が大きくなるとECNビットをセットする確率が下がるためであり、

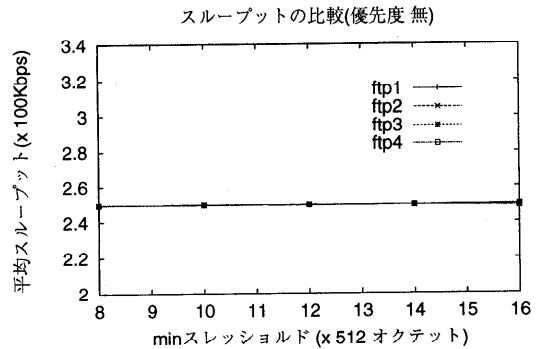


図6 min_{th} とスループットの比較 (優先度なし)
Fig. 6 Comparison of min_{th} and throughput (Non priority).

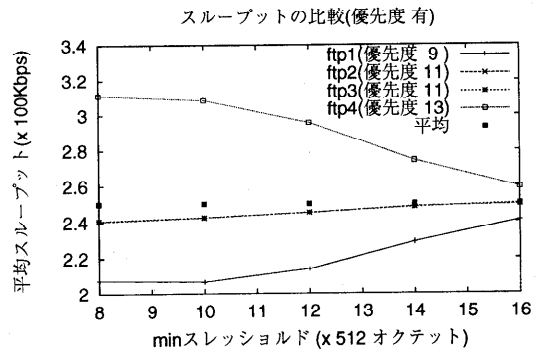


図7 min_{th} とスループットの比較 (優先度あり)
Fig. 7 Comparison of min_{th} and throughput (With priority).

非輻輳時には優先度の低いデータグラムの送信を制限しないことを示している。また、ftp2とftp3のスループットは240~250 Kbpsにおさまり、図6とほぼ同じ結果となった。さらに、全ftpの平均スループットはすべての min_{th} で250 Kbpsであり、図6の結果と同じ値である。したがって、優先制御により全体のスループットは低下しないことが分かった。

4.3 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512 オクテット、1024 オクテット、1536 オクテット、2048 オクテットと変化させた際のスループットの変化を比較する。図8に優先度を考慮しない場合を、図9に優先度を考慮した場合のシミュレーション結果を示す。

図8から、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。これは与えられた帯域を4つのコネクションで均等に利用しているからである。4つのftpの平均スループットは、データグラムサイズが小さい順に250 Kbps、249 Kbps、

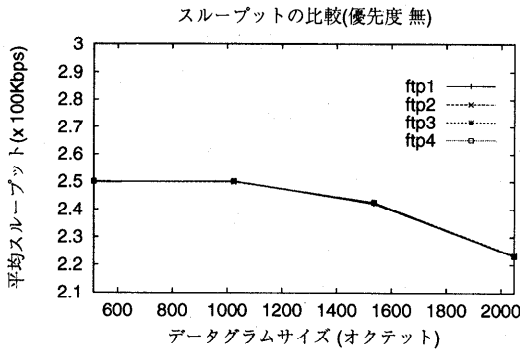


図 8 データグラムサイズとスループットの比較 (優先度なし)
Fig. 8 Comparison of datagram size and throughput (Non priority).

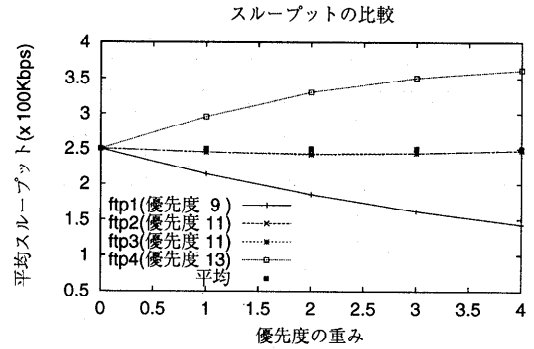


図 10 優先度の重みによるスループットの比較
Fig. 10 Comparison of throughput by weight of priority.

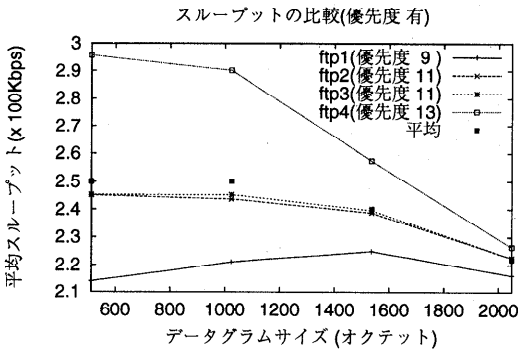


図 9 データグラムサイズとスループットの比較 (優先度あり)
Fig. 9 Comparison of datagram size and throughput (With priority).

242 Kbps, 223 Kbps となり、データグラムサイズが大きくなるに従い徐々に低下していることが分かる。これは、データグラムサイズが大きくなるほど、再送によるオーバーヘッドが大きくなるためである。しかし、文献 17) より TCP/IP の多くのデータグラムサイズは 512 オクテットであるため、ユーザはデータグラムサイズによる再送オーバーヘッドの相違を意識せずに優先度を割り当て、それにともなったスループットを得ることができる。

図 9 より、データグラムサイズを大きくするに従い、高優先のスループットは減少するが、低優先のそれはデータグラムサイズが 1536 オクテットまでは増加し、それ以降は減少している。ただし、データグラムサイズが大きくなるに従い、全体的に徐々に低下している。たとえば、データグラムサイズが 512 オクテットのとき、ftp1 と ftp4 のスループットはそれぞれ 214 Kbps, 296 Kbps であり、図 8 の結果 (250 Kbps) との比率はそれぞれ 85.6%, 118.4% となるのに対し、データグラ

ムサイズが 2048 オクテットでは 216 Kbps, 226 Kbps にとどまり、図 8 の結果 (223 Kbps) との比率もそれぞれ 96.9%, 101.3% である。この理由は、図 8 の結果と同様、再送オーバーヘッドによるものである。また、図 9 で 4 つの ftp の平均値は、データグラムサイズが小さい順に 250 Kbps, 250 Kbps, 240 Kbps, 222 Kbps であり、図 8 の各データグラムサイズにおけるスループットとほぼ同じ値である。したがって、優先制御による全体のスループットの低下はほとんど見られないことが分かる。

4.4 重みの変化による比較

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

図 10 では優先度の重み 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ということが確認できる。優先度の重みが $w = 0$ のときには優先制御を行わない場合であり、4 つの ftp で 250 Kbps のスループットが得られている。ftp1 は w の値が大きくなるに従ってスループットが低下しているのに対し、ftp4 は上昇している。たとえば、 $w = 2$ のときの ftp1, ftp4 の値はそれぞれ 185 Kbps, 330 Kbps となり、 $w = 4$ のときはそれぞれ 143 Kbps, 362 Kbps となっている。さらに、ftp2 と ftp3 は 240~250 Kbps であり、 $w = 0$ のときとスループットがほぼ等しい。また、各 ftp の平均スループットは $w = 0$ の場合と同じく 250 Kbps である。 w による ECN ビットセット確率の影響については 3 章で考察したが、 $w = 4$ において ftp1 のスループットは $w = 0$ のときの 42.8% の減少にとどまっておき、表 1 での確率が約 5 倍となったことを考慮すると下降幅は十分おさえられていることが確認された。ftp4 のスループットについても 44.8% の増加であり、同様のことがいえる。以上のことから、優先度の重み w の値を高く設定することに

表3 データの転送にかかる時間(秒)
Table 3 Translation time of data.

w	優先度 9		優先度 11		優先度 13	
	(1)	(2)	(1)	(2)	(1)	(2)
0	32.0	2.56	32.0	2.56	32.0	2.56
1	37.4	2.99	32.6	2.61	27.0	2.16
2	43.2	3.45	33.0	2.64	24.2	1.94
3	49.4	3.96	32.8	2.63	22.8	1.83
4	56.0	4.47	32.3	2.59	22.1	1.77

(1) : 1 Mbyte × 1 個

(2) : 80 byte × 1000 個

より、全体のスループットを低下させることなく、高いスループットを求めているユーザに対して適切なスループットを割当て可能なことが分かった。

次に、図 10 で得られたスループットで 1 Mbyte と 80 byte のファイル転送に何秒要するかを比較した結果を表 3 に示す。なお、表中の優先度 11 は ftp2 と ftp3 の平均スループットを用いており、各優先度の項目の左側に 1 Mbyte の転送に要する時間(秒)、右側に 80 byte のファイル 1000 個の転送に要する時間(秒)を表している。前者は通常のファイル転送を、後者はリアルタイム性の通信におけるファイル転送を想定できる。前者では優先制御を行わない場合 ($w = 0$) は 32.0 秒であり、優先度 9 では、 $w = 1, 4$ のとき、それぞれ 37.4 秒、56.0 秒となり、それぞれ 16.9%、75.0% の増加となる。後者の場合、 $w = 0$ のときは 2.56 秒であるのに対し、優先度 13 では $w = 1, 4$ においてそれぞれ 2.16 秒、1.77 秒となり、1 個あたり、0.40 ミリ秒、0.79 ミリ秒の減少となる。前者の増加は 2 倍以下にとどまることから、重要性の低いファイル転送を割り当てた場合では十分実用性に耐える。逆に後者の減少は体感的にはほとんど相違はないが、時間制約が厳しい場合はこの減少幅は大きく、その効果は前者の転送時間の増加を十分補うものである。

5. 今後の検討課題と結論

5.1 今後の検討課題

今回の実験では、ftp を用いたデータの転送による実験を行ったが、優先度の低い ftp としては、時間的制約がないデータを転送する場合を想定することができ、anonymous ftp での転送や、データのミラーリングが例としてあげられる。また、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が考えられる。他のプロトコルについては、対話性のない smtp や nntp の転送は、高いスループットを必要としないので優先度を低く設定することができ、逆に対話性のある telnet, http などの転送は優先度を高く設定

することが望まれる。

ところで、本方式では優先度や重みの設定はユーザが自由に行うことができるため、ユーザ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、偏りがな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。したがって、ユーザは得られるスループットの特性を理解したうえで、優先度を与える際に偏りが生じないように考慮する必要があるが、適切な優先度の値の基準を与えることは難しい。この制御方法とその評価については今後の課題であるが、1 つの解決法として、オペレーティングシステムのプロセスに優先度が与えられたときのタスクスケジューリングの方式を応用して、ユーザから指定された優先度をネットワークの状況に応じてシステムが変更する方策が考えられる。

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課題としたい。

5.2 結論

本論文では、明示的輻輳通知を用いて、優先度に応じた輻輳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。いずれの条件下でも、平均スループットは、優先制御を行わない場合と、優先制御を行った場合に、ほぼ同じ値になっているという結果が得られた。すなわち、RED ゲートウェイアルゴリズムの ECN ビットセット確率に変更を加えるだけで、平均スループットを落とすことなく、優先制御が実現可能であることを示した。また、重み w を変化させることによって優先制御の度合を調整できることを示した。

従来の IPv6 の優先度フィールドを用いる優先制御において、優先度の低いデータグラムは輻輳の条件下では廃棄されることを意図して設定されている¹⁵⁾。しかしながら、この方式では、廃棄することにより後に再送が必要となり、これによる再送オーバーヘッドが生じる。これに対して、本方式では、輻輳通知に RED ゲートウェイ方式を用い、前もってウィンドウサイズを下げるように通知する。これにより、優先制御のために廃棄されるデータグラムは減少し、オーバーヘッドを減らすという利点がある。

なお、本方式の優先制御は、IP ヘッダからデータ

グラムの優先度を取り出すことと、RED ゲートウェイアルゴリズムにより計算される ECN ビットセット確率 p_a の変更のみであり、これに要する処理時間は十分に小さいので、優先度処理によるオーバヘッドは考察の対象とはしなかった。ただし、本提案方式は、RED ゲートウェイ方式が実装されているルータを用いて実現する必要がある、実装されていないルータが介在するときは、これのオーバヘッドが加算される。

また、本方式により、バッファ長の閾値 (min_{th}) やデータグラムサイズ、優先度の重み w の変化に対する関係を、シミュレーションにより定量的に示したことは、今後の TCP の優先輻輳制御方式の設計に重要な知見を与える。

参考文献

- 1) Berners-Lee, T., Fielding, R. and Frystyk, H.: Hypertext Transfer Protocol - HTTP/1.0, RFC-1945 (1996).
- 2) Postel, J. and Reynolds, J.: Telnet Protocol Specification, RFC-854 (1983).
- 3) Postel, J. and Reynolds, J.: File Transfer Protocol (FTP), RFC-959 (1985).
- 4) Postel, J.: Simple Mail Transfer Protocol, RFC-821 (1982).
- 5) Kantor, B. and Lapsley, P.: Network News Transfer Protocol, RFC-977 (1986).
- 6) Postel, J.: Transmission Control Protocol, RFC-793 (1981).
- 7) Jacobson, V.: Congestion Avoidance and Control, *Proc. ACM SIGCOMM '88*, pp.314-329 (1988).
- 8) Newman, P.: Traffic Management for ATM Local Area Networks, *IEEE Communications Magazine*, Vol.32, No.8, pp.44-50 (1994).
- 9) Mankin, A. and Ramakrishnan, K.: Gateway Congestion Control Survey, RFC-1254 (1991).
- 10) Postel, J.: Internet Control Message Protocol, RFC-792 (1981).
- 11) Ramakrishnan, K. and Jain, R.: A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer, *Proc. ACM SIGCOMM '88*, pp.303-313 (1988).
- 12) Floyd, S.: TCP and Explicit Congestion Notification, *ACM Computer Communication Review*, Vol.24, No.5, pp.10-23 (1994).
- 13) Floyd, S. and Jacobson, V.: Random Early Detection Gateways for Congestion Avoid-

ance, *IEEE/ACM Transactions on Networking*, Vol.1, No.4, pp.397-413 (1993).

- 14) Braden, B., Clark, D., Crowcroft, J., et al.: Recommendations on Queue Management and Congestion Avoidance in the Internet, RFC-2309 (1998).
- 15) Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification, RFC-1883 (1995).
- 16) McCanne, S. and Floyd, S.: Network Simulator -ns (version2). <http://www-mash.cs.berkeley.edu/ns/>.
- 17) 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, pp.19-24 (1997).

(平成 10 年 5 月 8 日受付)

(平成 10 年 11 月 9 日採録)



上野 英俊

平成 9 年筑波大学第三学群情報学類卒業。現在、同大学院博士課程工学研究科在学中。次世代インターネットプロトコルの開発、性能評価、および、移動通信システムの研究等に従事。電子情報通信学会学生会員。



木村 成伴 (正会員)

平成 2 年東北大学工学部情報工学科卒業。平成 7 年同大学院博士後期課程修了。同年筑波大学電子・情報工学系講師。現在に至る。情報科学博士。代数プロセスの合成、フェアネスの導入等の研究に従事。電子情報通信学会、ソフトウェア科学会各会員。



海老原義彦 (正会員)

昭和 45 年東北大学工学部電子工学科卒業。昭和 50 年同大学院博士課程単位取得退学。同年同大学助手、現在、筑波大学電子・情報工学系教授。平成 10 年より同大学術情報処理センター長。工学博士。コンピュータネットワークアーキテクチャ、デジタル通信システムの性能評価、および知的通信システムの研究等に従事。電子情報通信学会会員。