

集約トラフィックに対する帯域制御方式と 利用可能帯域推定方式の提案

松田哲史^{†1‡2} 石原進^{‡2}

概要: ゲートウェイ装置(以下, GW)を介してベストエフォートサービスネットワークで接続される2拠点間で複数のアプリケーション(以下, APL)が同時に通信を行うとき, 2拠点間通信の利用可能帯域が変動するため, 高優先 APL の通信フローの受信レートが必要な通信帯域(以下, 必要帯域)未満となる問題(高優先通信帯域不足問題)が起き得る. この問題の対策として, GW が2拠点間通信の利用可能帯域を推定し, 得られた利用可能帯域推定値に基づきパケットスケジューラを用いて通信フロー毎の送信帯域制御を行うことを考える. この時, 利用可能帯域の推定値が推定誤差により真の利用可能帯域より大きくなる場合がある. このため, 真の利用可能帯域が高優先 APL の必要帯域より大きい場合でも, 高優先 APL の受信レートが必要帯域未満になることがある. 本論文では, 利用可能帯域推定処理から得られる情報を用いて受信レートの必要帯域に対する不足量を推定し, 推定した不足量を補うようにフロー毎の割当送信帯域を制御することで高優先通信帯域不足問題の発生頻度を減らす帯域制御方式を提案する. また, 利用可能帯域推定に必要な追加トラフィック量を従来手法より少なくできる利用可能帯域推定方式を提案する. 仮想マシンで構築した仮想ネットワーク環境で提案方式を実行し, 高優先通信帯域不足問題に対し, 提案方式が従来方式より高い抑制効果を示すことを確認した.

キーワード: 利用可能帯域推定, 帯域制御, 高優先通信帯域不足問題, 輻輳制御, パケットスケジューラ

1. はじめに

遠隔保守サービスシステムの様に, ゲートウェイ装置(以下, GW)を介してベストエフォートサービスネットワークで接続される2拠点間で複数のアプリケーション(以下, APL)が同時に通信を行う場合(図1)に, 2拠点間通信の利用可能帯域が変動するため, 高優先 APL の通信フロー(以下, フロー)の受信レートが必要な通信帯域(以下, 必要帯域)未満となることがある. 本論文ではこれを高優先通信帯域不足問題と呼ぶ. GW が複数フローからなる集約トラフィックの転送時にあたり, 個々のフローに対して帯域制御を行わなければ, GW を含むネットワークはフローの優先度に関係なくパケットの転送・廃棄処理を行う. 従って, 複数のフローの必要帯域の合計が利用可能帯域を超える場合, 高優先フローと低優先フローが優先度に関係なく廃棄され, 高優先通信帯域不足問題が起きる.

GW による集約トラフィック転送時にフローに対して帯域制御を行うための既存技術として, パケットスケジューラが存在する[1]. 帯域制御のためにパケットスケジューラを使うには, パケットスケジューラが送信するトラフィックの送信レートの総和の上限値(以下, 総レート)を決める必要がある. GW が実行するパケットスケジューラの総

レートが利用可能帯域より大きい場合, GW が転送したパケットはベストエフォートサービスネットワーク内のボトルネックリンクでフローの優先度に関係なく転送・廃棄されてしまう. 一方, GW が実行するパケットスケジューラの総レートが利用可能帯域より小さいと, 利用可能帯域を有効利用できない. このため, 利用可能帯域を GW が実行するパケットスケジューラの総レートとして使用することが望ましい. ただし, ベストエフォートサービスネットワークの利用可能帯域は動的に変化し, 利用可能帯域の値を直接知ることが通常できないため, 利用可能帯域を推定する必要がある. これらのことから, GW が集約トラフィックに対する帯域制御を行うために, GW が利用可能帯域を推定し, 得られた利用可能帯域推定値を総レートとしてパケットスケジューラを実行する方法が考えられる.

しかし, 利用可能帯域推定値は通常誤差を含み, 利用可能帯域推定値が真の利用可能帯域より大きくなる可能性がある. そのことが原因で以下の様な問題が生じる. 利用可能帯域推定値が真の利用可能帯域より大きいと, ベストエフォートサービスネットワーク内のボトルネックリンクで, 高優先フローのパケットが廃棄される可能性が高い. 高優先フローのパケットが廃棄されると, 当該フローの受信レートが, 廃棄パケット分だけ送信レートより低下する(図2). TCP フローの場合は, 輻輳制御動作[2]により, 送信元の送信レート自体が低下する可能性もある. これらのこと

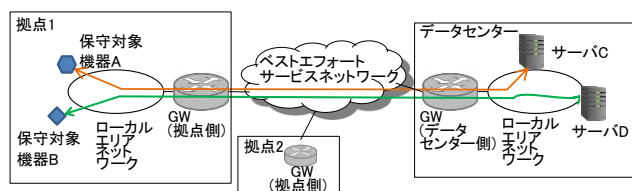


図1 遠隔保守サービスのシステム例

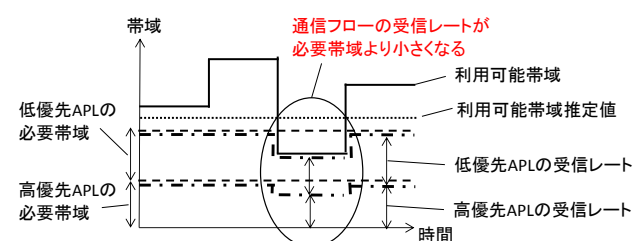


図2 高優先通信帯域不足問題の発生例

^{†1} 三菱電機(株)
Mitsubishi Electric Corporation.

^{‡2} 静岡大学
Shizuoka University

から、GW が利用可能帯域の推定処理とパケットスケジューリングを組み合わせて集約トラフィックの帯域制御を適切に行わないと、利用可能帯域が高優先フローの必要帯域より大きい時に、高優先フローに必要帯域分の送信帯域を割り当てても、高優先フローの受信レートが必要帯域未満となってしまう。

本論文では、GW が、利用可能帯域推定処理から得られる情報を用いて受信レートの必要帯域に対する不足量を推定し、推定した不足量を補うようにパケットスケジューラに設定する送信帯域を制御することで、高優先通信帯域不足問題の発生頻度を減らす帯域制御方式を提案する。また、利用可能帯域の推定方式として、送信側 GW が受信側 GW に受信パケット量を定周期で問い合わせ、送信パケット量と受信パケット量の比較によりパケットロス発生を検出することで、利用可能帯域推定に必要な追加トラフィック量を従来手法より少なくできる方式を提案する。そして、提案方式を仮想マシンで構築した模擬ネットワーク環境で実行して評価し、提案方式に高優先通信帯域不足問題抑制効果があることを示す。

論文の構成は以下の通りである。2 節で関連研究を述べ、3 節で提案方式を説明する。4 節で提案方式の評価結果を示し、5 節でまとめを述べる。

2. 関連研究

GW が利用可能帯域の推定を行い、複数の TCP もしくは非 TCP のフローからなる集約トラフィックのフロー単位の帯域制御を行う関連研究を整理する。複数フローからなる集約トラフィックのフロー単位の帯域制御を目的として GW が利用可能帯域の推定を行う手法が[3][4][5][6]で提案されている。以下ではこれらを、フローの帯域制御に関する記述内容、利用可能帯域推定のためのオーバーヘッド（アプリケーションの通信データに加えて送信が必要なデータ）の 2 つの観点で整理する。

Damjanovic らは[3]で、TCP Friendly Rate Control (TFRC) [7]の送信レート計算式を拡張した方式で利用可能帯域を推定する MulTFRC という方式を提案している。[3]はフローの帯域制御について、パケットスケジューラの使用を提案しているが、高優先通信帯域不足問題の対策について記述はない。[3]の利用可能帯域推定のためのオーバーヘッドは、TFRC と同等である。

TFRC の利用に伴うオーバーヘッドは、送信パケットに付加するシーケンス番号等のヘッダ情報と、受信側が RTT 時間に 1 回程度の頻度で送信するフィードバックパケットであり、送信パケット数に比例する。IETF 標準プロトコル DCCP[8]で TFRC を実装する場合、パケットに付与するヘッダ情報は 32byte であり、MTU 値 1500byte の場合、送信データの $32/1500 = 2\%$ 強がオーバーヘッドとなる。

Singh らは[4]で、2 拠点間の複数 TCP フローを対象とし

て、GW が複数本の TCP コネクションの輻輳制御ウィンドウを一括管理し、利用可能帯域を推定する Multi-Probe Aggregate TCP (MPAT)という方式を提案している。同方式は帯域制御について、TCP フローの送信帯域が指定比率になるようパケットスケジューリングするが、同論文には高優先通信帯域不足問題の対策について記述はない。[4]の利用可能帯域推定のためのオーバーヘッドは、TCP フローを対象とする場合は 0 で、非 TCP フローを対象とする場合は TFRC と同等である。

Subramanian らは[5]で、利用可能帯域を、複数本の TFRC コネクションの送信レートとして推定する OverQoS という方式を提案している。OverQoS はフローの帯域制御について、パケットスケジューラを適用し、利用可能帯域推定値が閾値未満になる確率を用いる確率的帯域保証方式である。高優先通信帯域不足問題の対策について記述はない。また、[5]の利用可能帯域推定のためのオーバーヘッドは、TFRC と同等である。

Balakrishnan らは[6]で、利用可能帯域を、送信パケットに付加するシーケンス番号等のヘッダ情報を用いて、TCP の輻輳制御方式に類似する方式で推定する Congestion Manager (CM)という方式を提案している。同方式ではフローの帯域制御について、フローの送信帯域が指定比率になるよう制御するパケットスケジューリング方法を用いている。高優先通信帯域不足問題の対策について記述はない。また、[6]の利用可能帯域推定のためのオーバーヘッドは、TFRC と同等である。

以上をまとめると、上述した手法では、高優先通信帯域不足問題の対策が示されておらず、利用可能帯域推定のオーバーヘッドは TFRC と同等である。前述した様に、TFRC のオーバーヘッドは送信 IP パケット数に比例して増える。しかし、一部の移動体通信サービスのように従量制課金契約の通信回線を利用する場合、オーバーヘッド削減が運用コスト削減につながるため、利用可能帯域推定のためのオーバーヘッドを削減することは有用な場合がある。

3. 提案方式

本節では、高優先通信帯域不足問題を改善する帯域制御方式を提案する。また、提案帯域制御方式の評価に使用する利用可能帯域推定方式として、利用可能帯域推定のためのオーバーヘッドを削減可能な利用可能帯域推定方式を提案する。

3.1 提案帯域制御方式

高優先通信帯域不足問題の発生頻度を減らすために、利用可能帯域推定処理から得られる情報を用いて受信レートの必要帯域に対する不足量を推定し、推定した不足量を補うように送信帯域を制御する帯域制御方式を提案する。最初に提案帯域制御方式の前提条件を示し、後の議論で用いる変数・パラメータを示す。次に、TCP の様に再送による

誤り回復を行うフローと、再送による誤り回復を行わないフローに分けて、それぞれについて帯域制御の考え方を示す。その後、フローの受信レートの必要帯域に対する不足量推定方法、受信レートの必要帯域に対する推定不足量を補うための送信帯域制御方法を説明する。

3.1.1 提案帯域制御方式の前提条件

まず、提案帯域制御方式の前提条件を説明する。提案帯域制御方式では、送信側 GW が利用可能帯域推定を行い、利用可能帯域推定値更新時に得られる情報を入力として、最低帯域保証型パケットスケジューラの総レートを利用可能帯域推定値に更新し、各フロー i の最低保証帯域値 $G_i(t)$ を更新する。利用可能帯域推定方式は、利用可能帯域推定値を更新する時に、新しい利用可能帯域推定値、集約トラフィックの受信レート、ならびにパケットロス発生有無の情報を提案帯域制御方式に提供可能であるものとする(図3)。

[3][4][5][6]に示される利用可能帯域推定方式はこの前提条件を満たしており、これは特殊な前提条件ではない。

帯域制御対象のフローの前提条件を説明する。各フローは優先度の降順に番号を付けられているものとする。再送による誤り回復を行うフロー i ($i \in F_r$)については、アプリケーションの要件から、平均受信レートが必要帯域以上か否かを判定するための期間が設定されているものとする。この期間を、平均受信レート算出期間と呼ぶ。再送による誤り回復を行わないフロー i ($i \in F_u$)については、APL がパケットロスの発生を想定して作られており、FEC による誤り回復用パケット[9]などの冗長パケットを送信し、必要帯域より大きい送信レートでデータを送信するものとする。

以下で提案帯域制御方式の説明に用いる変数を表1に示し、提案帯域制御方式の設定パラメータを表2に示す。

3.1.2 再送による誤り回復を行うフローに対する帯域制御方式

再送による誤り回復を行うフロー i の帯域制御に使用する最低保証帯域値 $G_i(t)$ の求め方を説明する。再送による誤り回復を行うフローは、ある時点で送信できなかったデータを後で送信することが可能である。そこで、フロー i の受信レートの必要帯域に対する不足量推定値 $\delta_i(t)$ の積算値 $\Delta_i(t)$ を求め、優先度が高いフローから順に $\Delta_i(t)$ が0以下になる様に $G_i(t)$ を決める。こうすることで、優先度が高いフローほど、複数回の利用可能帯域推定値更新にわたる平均受信レートが必要帯域 $R_f(i)$ 以上になる可能性が高くなる。

$\delta_i(t)$ を以下の様に推定する。再送による誤り回復を行うフローでは、パケットロス発生時に、パケットロスにより受信レートが送信レートの $R(t-1)/W(t-1)$ 倍に低下するのに加えて、輻輳制御が働き送信元の送信レートが変わる可能性がある。このため、送信元の平均的な送信レートは、パケットスケジューラが集約トラフィック内の誤り回復を行う複数のフローに割り当てた送信帯域の合計値を、

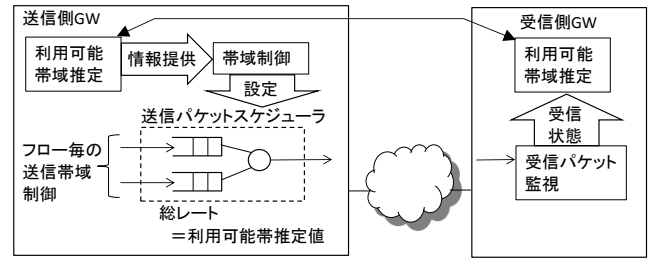


図3 提案帯域制御方式が前提とする構成

表1 提案制御方式の説明に用いる変数

変数	説明
$G_i(t)$	t 回目の利用可能帯域推定値更新時に、フロー i に対して設定するパケットスケジューラの最低保証帯域値。
$\delta_i(t)$	t 回目の利用可能帯域推定値更新時における、再送による誤り回復を行うフロー i の受信レートの必要帯域に対する不足量推定値。
$\Delta_i(t)$	再送による誤り回復を行うフロー i について、 $\delta_i(t)$ を直近 C 回の利用可能帯域推定値更新時分積算した値。
$W(t)$	t 回目の利用可能帯域推定値更新時に求める利用可能帯域推定値。
$R(t-1)$	t 回目の利用可能帯域推定値更新時に測定する、送信レート $W(t-1)$ で送信した集約トラフィックの受信レート。
$s(t)$	t 回目の利用可能帯域推定値更新時に求める、集約トラフィックの受信レートと送信レートの比。
$H(t)$	t 回目の利用可能帯域推定値更新時における、直近 M 回の利用可能帯域推定値更新時の $s(t)$ の履歴。 $\{s(t), s(t-1), \dots, s(t-(M-1))\}$
$s_p(t+1)$	t 回目の利用可能帯域推定値更新時に求める、 $s(t+1)$ の値の下限の予測値。 $H(t)$ を入力に使用して求める。

表2 提案帯域制御方式が使用する設定パラメータ

パラメータ	説明
F_r	再送による誤り回復を行うフローの集合。
F_u	再送による誤り回復を行わないフローの集合。
F	帯域制御対象のフローの集合。 $F = F_r \cup F_u$ 。
C	再送による誤り回復を行うフローについて、フローの受信レートの必要帯域に対する不足量の積算対象とする利用可能帯域推定値更新回数。
M	$H(t)$ に記録する直近の $s(t)$ の数。
$R_f(i)$	i 番目のフローの必要帯域。($1 \leq i \leq N_f$)

誤り回復を行うフローに等分配した値になると見込まれる。この影響を考慮した受信レートの最小値と、フローの必要帯域との差を $\delta_i(t)$ とする。以上より、 t 回目の利用可能帯域推定値更新時に、再送による誤り回復を行うフロー i に対する受信レートの必要帯域に対する不足量推定値を以下の式で計算する。

$$\delta_i(t) = \begin{cases} R_f(i) - \left(\frac{R(t-1)}{W(t-1)} \right) \min \left(G_i(t-1), \left(\frac{\sum_{j \in F_r} G_j(t-1)}{|F_r|} \right) \right) & (\text{パケロス有}) \\ R_f(i) - G_i(t-1) & (\text{パケロス無し}) \end{cases}$$

$$\Delta_i(t) = \sum_{k=t-C}^t \delta_i(k)$$

C の値は、アプリケーションの要件による平均受信レート算出期間と、利用可能帯域推定値更新頻度に基づき決める。

受信レートの必要帯域に対する推定不足量を補うため、 $\sum_{i=1}^{|F|} G_i(t) \leq W(t)$ を満たす範囲で、優先度が高いフローから順に $\Delta_i(t) \leq 0$ になる様に $G_i(t)$ を決める。ただし、フローの最低保証帯域値が短時間に大きく変動すると、TCPの輻

輻輳制御動作に悪影響を与えることが懸念されるので、必要帯域 $R_f(i)$ に対する最低保証帯域値の加算量は B_{lim} を上限とする。以上より、 t 回目の利用可能帯域推定値更新時に、再送による誤り回復を行うフロー i の $G_i(t)$ を、優先度が高いフローから順に、以下の式で更新する。

$$G_i(t) = \begin{cases} \min(W(t) - \sum_{j=1}^{i-1} G_j(t), \min(R_f(i) + B_{lim}, R_f(i) + \Delta_i(t))) & (\Delta_i(t) > 0) \\ \min(W(t) - \sum_{j=1}^{i-1} G_j(t), R_f(i)) & (\Delta_i(t) \leq 0) \end{cases}$$

3.1.3 再送による誤り回復を行わないフローに対する帯域制御方式

次に、再送による誤り回復を行わないフロー i の帯域制御に使用する最低保証帯域値 $G_i(t)$ の求め方を説明する。再送による誤り回復を行わないフローは、FEC による誤り回復を想定すると、各時点の受信レートが必要帯域以上になるように制御することが必要となる。 t 回目の利用可能帯域推定値更新時に、パケットロスがある場合は、 $W(t-1)$ の送信レートで送信した集約トラフィックの受信レートがパケットロスで $R(t-1)$ になったとみなして、各フローが同じ割合でロスしたと仮定すると、再送による誤り回復を行わないフローの受信レートは、送信レートの $s(t) = R(t-1)/W(t-1)$ 倍となる。パケットロスが無い場合はフローの受信レートは送信レートに等しくなる ($s(t) = 1$ 倍)。 t 回目の利用可能帯域推定値更新で求めた $W(t)$ に対する $R(t)/W(t)$ の値はクロストラフィックの量に依存し、 t 回目の利用可能帯域推定値更新時には分からないが、クロストラフィック量の変動に関する仮定をおけば、 $s(t+1)$ の取り得る値の下限の予測値 $s_p(t+1)$ を求めることはできる。

3.1.1 で述べたフローについての前提より、 $\sum_{i=1}^{|F|} G_i(t) \leq W(t)$ を満たす範囲で、優先度が高いフローから順に、再送による誤り回復を行わないフロー i の最低保証帯域値を必要帯域の $1/s_p(t+1)$ 倍に設定することで、優先度が高いフローほど受信レートが必要帯域以上になる可能性が高くなる。

$s(t+1)$ の取り得る値の下限の予測値 $s_p(t+1)$ を以下の方法で予測する。予測の前提条件として、クロストラフィックの変動が、直近 M 回の利用可能帯域推定値更新時の変動と同程度であると仮定する。この仮定に基づき、 $s_p(t+1)$ の値を、 $H(t)$ に記録される直近 M 回の $s(t)$ の値を用いて $\min(s(t), \dots, s(t-(M-1)))$ とする。 $H(t)$ を用いて $s_p(t+1)$ を推測する方法に、ARIMA モデルなどの時系列解析の手法[10]を適用することも考えられるが、本論文では、クロストラフィックモデルに依存せず、処理が簡単であるため、前記の方法を用いることとした。 M の値は、クロストラフィックの変化周期として想定する時間長と、利用可能帯域推定値更新頻度に基づき決める。

以上より、受信レートが必要帯域未満となる可能性を低くするため、再送による誤り回復を行わないフロー i の最低保証帯域値 $G_i(t)$ を以下の様に決める。送信側 GW が、 $\sum_{i=1}^{|F|} G_i(t) \leq W(t)$ を満たす範囲で、優先度が高いフローか

ら順に、再送による誤り回復を行わないフロー i の最低保証帯域値を必要帯域の $1/s_p(t+1)$ 倍に設定する。すなわち、利用可能帯域推定値更新時に、再送による誤り回復を行わないフロー i に対する最低保証帯域値 $G_i(t)$ を、優先度が高いフローから順に、以下の式で更新する。

$$G_i(t) = \min(W(t) - \sum_{j=1}^{i-1} G_j(t), R_f(i)/s_p(t+1))$$

3.2 提案利用可能帯域推定方式

提案帯域制御方式の評価に使用する利用可能帯域推定方式として、利用可能帯域推定のためのオーバーヘッドが小さい利用可能帯域推定方式を提案する。骨子は以下の通りである。

- ・送信側 GW が受信側 GW で管理する受信パケット数情報をメッセージで問い合わせ、自ら管理する送信パケット数情報と比較することでパケットロス検出を行う。
- ・パケットロス検出結果に基づき、TCP の輻輳制御方式同様の Additive Increase Multiplicative Decrease (AIMD) の考え方で利用可能帯域推定値を更新する。

同方式の説明に使用する変数を表 3 に、提案帯域制御方式の設定パラメータを表 4 に示す。

3.2.1 提案利用可能帯域推定方式の処理内容

提案利用可能帯域推定方式は、(i) 送信側 GW が T_c を周期として受信側 GW で管理する受信パケット数情報をメッセージで問い合わせ、パケットロス検出に必要な情報を取得する処理、(ii) 送信側 GW が利用可能帯域推定値を更新する処理の 2 段階から成る。図 4 に、送信側 GW が t 番目の利用可能帯域推定において、受信側 GW からパケットロス検出に必要な情報を取得する手順を示す。図 4 において送信側 GW は、①で自身の $P_s(t)$ と $T_s(t)$ を取得し、④で受信側 GW から $P_r(t)$ 、 $B_r(t)$ 、 $T_r(t)$ を取得する。

ベストエフォートサービスネットワークは優先制御を行わないため、パケット転送経路変化時を除いて、受信側 GW は、送信側 GW が送信したパケットを送信順で受信する。よって、送信側 GW が送信するパケットと受信パケット量読み出し要求の前後関係は定常的には入れ替わらない。従って、

表 3 提案利用可能推定方式の使用変数

変数	説明
$P_s(t)$	t 回目の利用可能帯域推定時に送信側 GW が記録する累積送信パケット数。
$T_s(t)$	t 回目の利用可能帯域推定時に送信側 GW が累積送信パケット数を読み出した時刻。
$P_r(t)$	t 回目の利用可能帯域推定時に受信側 GW が読み出す累積受信パケット数。
$B_r(t)$	t 回目の利用可能帯域推定時に受信側 GW が読み出す累積受信バイト数。
$T_r(t)$	t 回目の利用可能帯域推定時に受信側 GW が累積受信パケット数を読み出した時刻。
$L(t)$	t 回目の利用可能帯域推定時に求めるパケットロス数。
$R(t-1)$	t 回目の利用可能帯域推定時に測定する、送信レート $W(t-1)$ で送信した集約トラフィックの受信レート。
$W(t)$	t 回目の利用可能帯域推定で求める利用可能帯域推定値。
$D(t)$	t 回目の利用可能帯域推定で求める、利用可能帯域推定値増加時の加算量。

表 4 提案利用可能帯域推定方式の設定パラメータ

パラメータ	説明
T_c	送信側 GW が受信パケット量読み出し要求を送信する周期.
θ_H, θ_L	利用可能帯域推定値が過大か否かを判断するために使用するパケットロス数の閾値. $\theta_H \geq \theta_L$.
D_{ini}	利用可能帯域推定値増加時の加算量の初期値.
D_{max}	利用可能帯域推定値増加時の加算量の上限値.
α	利用可能帯域推定値増加時の加算値を増やすために使用する乗数. $\alpha > 1$ とする.
β	パケットロス数 $> \theta_H$ の場合に, 新たな利用可能帯域推定値を求めるため, R から減算する値.
W_{ini}	利用可能帯域推定値の初期値.

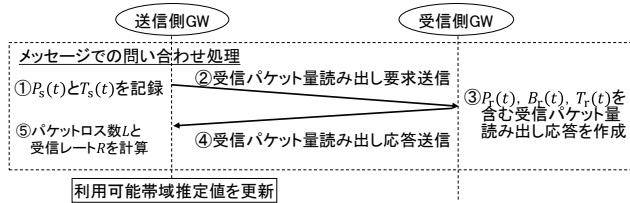


図 4 パケットロス検出に必要な情報の取得手順

図 4 の⑤で $L(t)$ と $R(t-1)$ は以下の式で計算できる.

$$L(t) = (P_s(t) - P_s(t-1)) - (P_r(t) - P_r(t-1))$$

$$R(t-1) = (B_r(t) - B_r(t-1)) / (T_r(t) - T_r(t-1))$$

送信側 GW は, パケットロス検出結果と利用可能帯域推定値の増加量を表す変数 $D(t)$ を使用して, AIMD の考え方に基づいて, 新しい利用可能帯域推定値 $W(t)$ を求める. 利用可能帯域推定値が過大か否かを L の値で判断するための閾値として θ_H, θ_L の 2 つを使用する. $L > \theta_H$ の場合は利用可能帯域推定値が過大と判断して, Multiplicative Decrease のため受信レートを元に

$$W(t) = R(t-1) - \beta$$

とする. $R(t-1)$ から β を減算する理由は, ボトルネックリンクのバッファに空きを作り, パケットロス発生を止めるためである. $\theta_H \geq L > \theta_L$ の場合は利用可能帯域推定値が少し大きいと判断して,

$$W(t) = (W(t-1) + R(t-1)) / 2$$

とする. $\theta_L \geq L$ の場合は利用可能帯域推定値が小さいと判断して,

$$W(t) = W(t-1) + D(t-1)$$

とする. この場合, $D(t)$ の値は, TCP の輻輳制御方式の Slow Start フェーズでの指数関数的増加と, その後の Additive Increase と同様の動作になる様に,

$$D(t) = \min(\alpha D(t-1), D_{max})$$

とする. $L > \theta_L$ の場合は $D(t) = D_{ini}$ とする. 初期値については, $W(1) = W_{ini}$, $D(1) = D_{ini}$ とする.

3.2.2 提案利用可能帯域推定方式のオーバーヘッド

提案方式のオーバーヘッドは, 利用可能帯域推定実行時に送信側 GW が受信側 GW 行う受信パケット量読み出し要求/応答メッセージであり, 送信パケット数に依存せず利用可能帯域推定の実行頻度に比例する. 受信パケット量読み

出し要求は, メッセージ種別情報(2byte)と, 応答との対応付けのためのシーケンス番号情報(2byte)を含めば良いので, UDP/IPv4 パケットで送信する場合で 32byte のサイズの IP パケットで実現可能である. 受信パケット量読み出し応答は, メッセージ種別情報(2byte), 要求との対応付けのためのシーケンス番号情報(2byte), 受信パケット数情報(8byte), 受信バイト数情報(8byte)と時刻情報(8byte)を含めば良いので, UDP/IPv4 パケットで送信する場合で 56byte のサイズの IP パケットで実現可能である. よって, 利用可能帯域推定のためのオーバーヘッドは, 1 回の利用可能帯域推定につき 88byte であり, 1 秒に 1 回利用可能帯域推定を実行する場合で 0.704kbps となる.

2 節で説明したように, 既存研究で TFRC ベースの方式で例えば 1Mbps の通信量があった場合, 通信量の 2%強 (20kbps) が利用可能帯域推定のオーバーヘッドとなり, 提案方式のオーバーヘッドの方が小さい. ただし, 提案利用可能帯域推定方式は T_c の周期で利用可能帯域推定値を更新し, T_c はラウンドトリップタイムより十分大きい値にする必要があるため, ラウンドトリップタイム毎に利用可能帯域推定値を更新する既存方式よりも, 利用可能帯域変化への追従性は劣る.

4. 提案方式の評価結果

4.1 評価項目と評価環境

本節では, 提案方式の評価項目と評価環境を説明する. 提案方式について, 高優先通信帯域不足問題抑制効果の従来方式との違いと, 利用可能帯域変化への追従性の確認を目的に評価にした. 評価は, 仮想マシンで構築した模擬ネットワーク環境で提案方式を実行して行った. 評価項目と評価方法を表 5 に示す.

8 コア/8GB メモリの PC で実行する Oracle VirtualBox 上で 6 台の Linux OS 仮想マシン(以下, VM)を内部ネットワ

表 5 評価項目と評価方法

評価項目	評価方法
1) 高優先通信帯域不足問題抑制効果	複数 UDP クロストラフィックを周期的に発生させ, 帯域制御対象の 5 本の TCP フローもしくは UDP フローを流し, 各フローの受信レートを測定.
2) 利用可能帯域変化への追従性	複数 UDP クロストラフィックをランダムに発生させ, 帯域制御対象の 5 本の TCP ないしは UDP フローを流し, 利用可能帯域推定値と, 各フローの受信レートを測定.

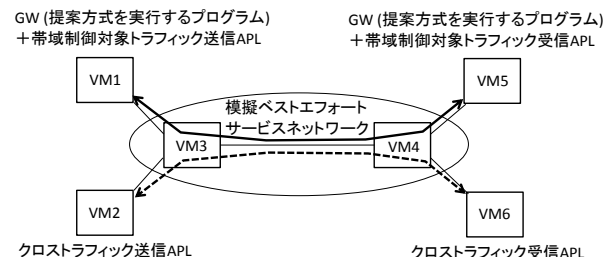


図 5 評価用の仮想マシン構成

表 6 評価で利用した提案方式のパラメータ

パラメータ	値	パラメータ	値	パラメータ	値	パラメータ	値
T_c	1.0 秒	D_{ini}	50kbps	β	100kbps	C	10
θ_H	1	D_{max}	1Mbps	W_{ini}	1Mbps	M	10
θ_L	0	α	1.2			B_{lim}	50kbps

ークで接続し、評価用に使用した。提案方式の実行プログラムの他、帯域制御対象トラフィック、クロストラフィックの送受信のために iperf3 [11]を VM 上で実行した。TCP の輻輳制御方式は Cubic を使用した。最低帯域保証型パケットスケジューラとして Linux の htb [12]を使用した。htb でフローの最低保証帯域値に 0 を設定できないため、最低保証帯域値は 80kbps 以上に設定した。

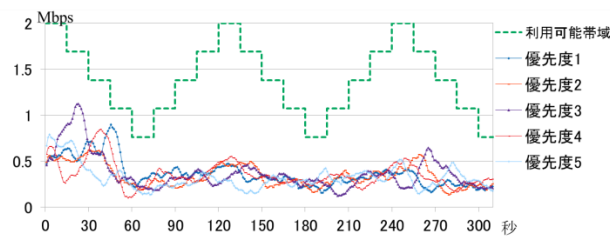
評価で利用した仮想ネットワークの構成を図 5 に示す。ベストエフォートサービスネットワークのボトルネックリンクと遅延を模擬するため、VM3 と VM4 の間の双方向の packets 送信に対して 5Mbps でシェーピングし、片道 25ms の遅延時間を付加した。

評価に使用した、提案利用可能帯域推定方式と提案帯域制御方式のパラメータを表 6 に示す。表 6 の α と β の値は、 α は 1.1 と 1.2 の 2 通り、 β は 150kbps と 100kbps の 2 通りを組み合わせ、4 通りの α と β の値で利用可能帯域推定動作と高優先通信帯域不足問題抑制効果を評価した結果に基づき選定した。

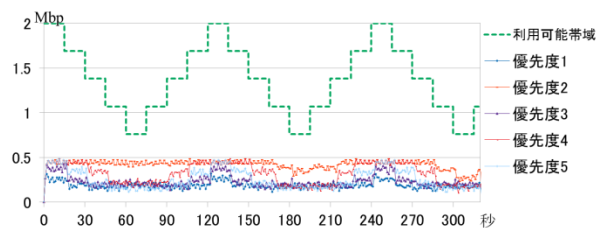
4.2 高優先通信帯域不足問題の抑制効果

本節では、GW が何も帯域を制御しない場合、従来の帯域制御方式、ならびに提案帯域制御方式の高優先通信帯域不足問題抑制効果を、制御対象フローの受信レートが必要帯域未達となる割合を指標に用いて比較する。高優先通信帯域不足問題が抑制されている場合、利用可能帯域が減る時に優先度が低いフローから順に受信レートが減り、利用可能帯域が増える時に優先度が高いフローから順に受信レートが増えることが期待される。こうした状況を観察するため、クロストラフィックとして常に送信する 3Mbps CBR UDP フロー 1 本、周期的に送信を開始/停止する 310kbps CBR UDP フロー 3 本を使用した。後者のフローの周期を調整して、利用可能帯域を 15 秒毎に 2Mbps→1.69Mbps→1.38Mbps→1.07Mbps→760kbps→1.07Mbps→1.38Mbps→1.69Mbps→2Mbps と変化させた。帯域制御対象のフローは 5 本で 320 秒間送信し、各フローの必要帯域を 400kbps に設定した。5 本の異なる優先度の TCP フローの場合、5 本の異なる優先度の CBR UDP フローの場合の 2 通りで評価した。TCP、UDP いずれの場合も、優先度 1 が最も優先度が高く、優先度の数字が増えるに従い優先度が低くなる設定とした。TCP フローの 10 秒平均受信レートと、UDP フローの 1 秒間の受信レートを、Linux の conntrack エントリからフロー毎の受信バイト数を 1 秒毎に読み出し求めた。

最初に、利用可能帯域推定と帯域制御を行わない方式に



(a) TCP フローの受信レート



(b) UDP フローの受信レート

図 6 帯域制御をしない方式の受信レート

表 7 受信レートが必要帯域未達となる割合(帯域制御なし)

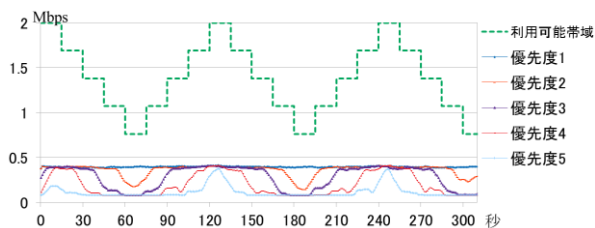
	優先度 1 (400kbps)	優先度 2 (400kbps)	優先度 3 (400kbps)	優先度 4 (400kbps)	優先度 5 (400kbps)
TCP	75.7%	69.6%	75.7%	73.9%	86.1%
UDP	100%	20.3%	97.2%	62.1%	87.9%

についての測定結果を示す。TCP フローの 10 秒平均受信レートと UDP フローの受信レートを図 6(a),(b)に、最初と最後の 10 秒間を除く期間において、5 本の TCP フローの 10 秒平均受信レートと 5 本の UDP フローの受信レートが必要帯域未達となる割合を表 7 に示す。

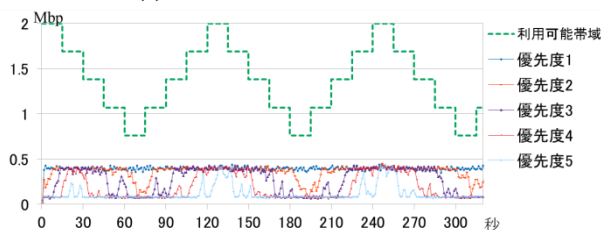
TCP フローについては、優先度と無関係に 5 本のフローが同じ様な受信レートになっており、受信レートが必要帯域未達となる割合も 5 本の TCP フローで 69%以上の同じ様な値になっている。UDP フローについては、優先度 2 と優先度 4 のフローが他のフローに比べて受信レートが必要帯域未達となる割合が低くなっている。TCP、UDP いずれの場合も、優先度が高いフローほど、受信レートが必要帯域未達となる割合が低いという関係にはなっておらず、受信レートが必要帯域未達となる割合も、優先度 2 の UDP フローを除くといずれも 60%より大きい。このように、GW が何も帯域を制御をしない場合に、高優先通信帯域不足問題が起きることが確認できた。

必要帯域を最低保証帯域値に設定する従来の帯域制御方式について、利用可能帯域推定に本論文で提案する方式を用いた場合の測定結果を示す。TCP フローの 10 秒平均受信レートと UDP フローの受信レートを図 7 にしめす。最初と最後の 10 秒間を除く期間で、5 本の TCP フローの 10 秒平均受信レートと 5 本の UDP フローの受信レートが必要帯域未達となる割合を表 8 に示す。

TCP、UDP どちらでも、利用可能帯域が減少する時は優先度が低いフローから受信レートが下がり、利用可能帯域が増加する時は優先度が高いフローから受信レートが上がる動作となっている。受信レートが必要帯域未達となる割



(a) TCP フローの受信レート



(b) UDP フローの受信レート

図7 従来型の方式の受信レート

表8 受信レートが必要帯域未満となる割合(従来型方式)

	優先度1 (400kbps)	優先度2 (400kbps)	優先度3 (400kbps)	優先度4 (400kbps)	優先度5 (400kbps)
TCP	82.1%	87.9%	90.7%	94.3%	100%
UDP	60.7%	70%	79%	91%	97.6%

合は、優先度が高いフローほど低くなっている。従って、必要帯域を最低保証帯域値に設定する方式には、高優先通信帯域不足問題の抑制効果があるといえる。

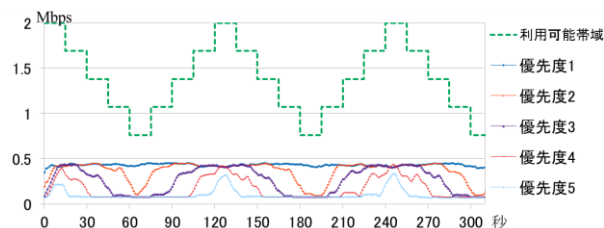
次に提案方式での測定結果を図8と表9に示す。図8に示すTCPフローとUDPフローの受信レートの変化は、利用可能帯域が減少する時は優先度が低いフローから受信レートが下がり、利用可能帯域が増加する時は優先度が高いフローから受信レートが上がる動作となっている。表9に示す受信レートが必要帯域未満となる割合は、優先度が高いフローほど低くなっている。従って、必要帯域を最低保証帯域値に設定する方式に、高優先通信帯域不足問題の抑制効果がある。

従来方式の評価結果(表8)を、提案方式の評価結果(表9)と比較すると、提案方式の方が優先度1から4のフローで受信レートが必要帯域未満となる割合が低く、提案方式の方が従来方式より高優先通信帯域不足問題の抑制効果が高いことが分かる。但し、提案方式は優先度が高いフローに必要な帯域より大きい帯域を割り当てることがあるため、必要帯域を最低保証帯域値に設定する従来方式に比べて、優先度が低いフローに割り当てられる帯域が減る副作用がある。

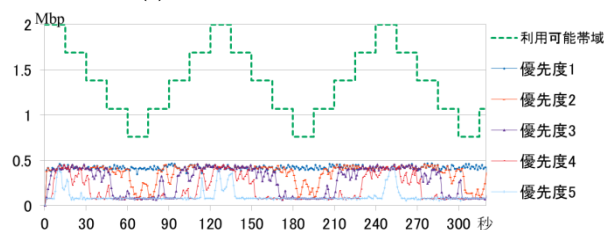
以上より、提案帯域制御方式が、従来方式より高い高優先通信帯域不足問題の抑制効果を持つことが確かめられた。

4.3 利用可能帯域変化への提案方式の追従性

提案方式の利用可能帯域変化への追従性を評価するため、利用可能帯域がランダムに変化する環境で、利用可能帯域推定値と、制御対象フローの受信レートを測定した結果を示す。実際のネットワークで利用可能帯域がランダム



(a) TCP フローの受信レート



(b) UDP フローの受信レート

図8 提案方式の受信レート

表9 受信レートが必要帯域未満となる割合(提案方法)

	優先度1 (400kbps)	優先度2 (400kbps)	優先度3 (400kbps)	優先度4 (400kbps)	優先度5 (400kbps)
TCP	0%	32.5%	67.9%	96.1%	100%
UDP	14.5%	37.6%	63.4%	88.3%	98.6%

変化する状況を模擬するために、クロストラフィックとして、常に送信する3Mbps CBR UDPフロー1つと、通信継続時間が指数分布近似の100kbps CBR UDPフローを、ポアソン生起を近似するように発生させ、最大で4本の100kbps CBR UDPフローが発生するようにした。帯域制御対象のフローは5本で320秒間送信し、優先度が高いフロー(優先度1)から順に必要な帯域を300kbps, 400kbps, 500kbps, 200kbps, 600kbpsに設定し、5本の異なる優先度のTCPフローがある場合、および5本の異なる優先度のCBR UDPフローがある場合について調べた。

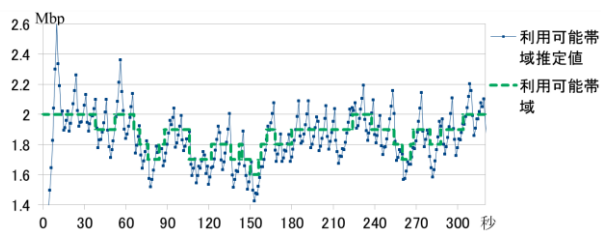
帯域制御対象のフローがTCPとUDPの場合の利用可能帯域推定値の変化を図9、受信レートの変化を図10、受信レートが必要帯域未満となる割合を表10に示す。

図9の結果より、帯域制御対象のフローがTCP, UDPのいずれでも、100kbpsから200kbpsの利用可能帯域の変動に対して、利用可能帯域推定値は3秒から5秒の遅れと最大400kbps程度のオーバーシュート/アンダーシュートで追従していることが分かる。

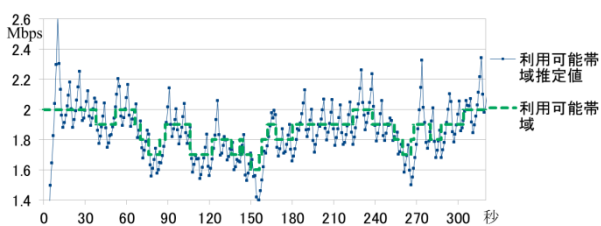
フローの受信レートについては、本評価シナリオでは利用可能帯域の最小値が1600kbpsなので、優先度1から優先度4のフローの受信レートは必要帯域以上の値になることが期待される動作となる。図10では、利用可能帯域推定値が1400kbps程度に落ち込む140秒から160秒の間を除くと、優先度1から優先度4のフローの受信レートは必要帯域に程度の値になっている。但し、優先度1から3のフローの受信レートが必要帯域より大きくなる時が多く、その分、特に優先度5のフローの受信レートが低くなっている。表10を見ると、最も優先度が高い優先度1のTCPフローに

ついて、受信レートが必要帯域未満となる割合が 34%と高くなっている。優先度 1 の TCP フローの 10 秒間の平均受信レートの最小値は 294kbps なので、平均受信レートが 34%の割合で 300kbps を数 kbps 下回っていることとなる。これは、提案方式の処理内容から、必要帯域が小さい TCP フローの提案帯域制御方式の変数 $\Delta_i(t)$ が、必要帯域が大きい TCP フローの $\Delta_j(t)$ より小さくなり、最低保証帯域値 $G_i(t)$ が必要帯域より大きい値になる機会が相対的に少ないことが理由と考えられる。

以上より、提案利用可能帯域推定方式は、100kbps から 200kbps の利用可能帯域の変動に 3 秒から 5 秒の遅れで追従可能であり、利用可能帯域変動時に提案帯域制御方式に

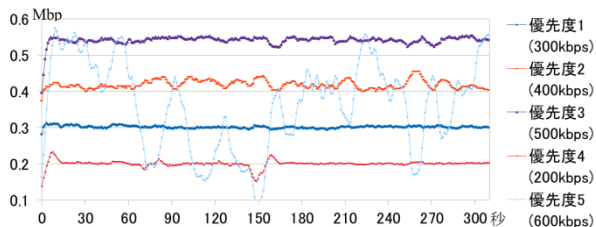


(a) 帯域制御対象のフローが TCP

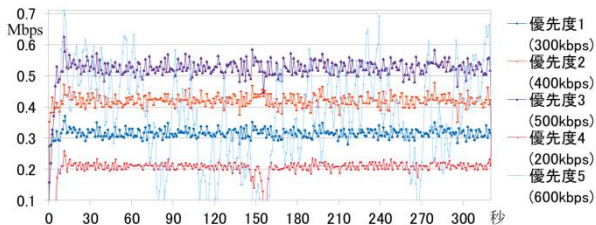


(b) 帯域制御対象のフローが UDP

図 9 利用可能帯域推定値の変化



(a) TCP フローの受信レート



(b) UDP フローの受信レート

図 10 フローの受信レート

表 10 受信レートが必要帯域未満となる割合
 (提案方式の追従性の評価)

	優先度 1 (300kbps)	優先度 2 (400kbps)	優先度 3 (500kbps)	優先度 4 (200kbps)	優先度 5 (600kbps)
TCP	34%	0.4%	0%	30.5%	100%
UDP	9.6%	12%	8.9%	28.2%	95.5%

よる高優先通信帯域不足問題の抑制効果が得られることを確認できた。

5. おわりに

本論文では、GW で接続される 2 拠点間の利用可能帯域が変動する場合に高優先フローの受信レートが必要帯域を下回る高優先通信帯域不足問題を対象として、フローの送信帯域制御方式を提案した。提案方式は、利用可能帯域推定処理から得られる情報を用いて受信レートの必要帯域に対する不足量を推定し、推定した不足量を補うように送信側で割り当てる送信帯域を制御することで高優先通信帯域不足問題の発生頻度を減らす。仮想ネットワーク環境上の実装評価により、提案手法が既存手法より高優先通信帯域不足問題の抑制効果が高いことを示した。加えて、提案する帯域制御方式の評価に使用する利用可能帯域推定方式として、利用可能帯域推定に必要な追加トラフィック量を従来手法より少なくできる利用可能帯域推定方式を提案し、仮想ネットワーク環境上の実装評価により、利用可能帯域推定値更新周期が 1 秒の場合に 100kbps から 200kbps の利用可能帯域の変動に 3 秒から 5 秒の遅れで追従可能であることを示した。帯域制御方式における受信レートの必要帯域に対する不足量の推定方法の改善と、利用可能帯域推定方式の追従性改善が今後の課題である。

参考文献

- [1] <https://www.tldp.org/HOWTO/Adv-Routing-HOWTO/lartc.qdisc.classful.html> (参照 2018-07-04).
- [2] A. Afanasyev, N. Tilley, P. Reiher, L. Kleinrock, "Host-to-Host Congestion Control for TCP", IEEE Communications Surveys & Tutorials, pp.304-342, 2010.
- [3] D. Damjanovic and M. Welzl, "An extension of the TCP steady-state throughput equation for parallel flows and its application in MultFRC", IEEE/ACM Trans. Netw., vol. 19, no. 6, pp. 1676-1689, Dec. 2011.
- [4] M. Singh, P. Pradhan, and P. Francis, "MPAT: Aggregate TCP congestion management as a building block for internet QoS" in Proc. ICNP 2004, pp. 129-138, 2004.
- [5] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz, "OverQoS: An overlay based architecture for enhancing Internet QoS," in Proc. NSDI 2004, pp. 11-16, 2004.
- [6] H. Balakrishnan, H. S. Rahul and S. Seshan, An Integrated Congestion Management Architecture for Internet Hosts, in Proc. SIGCOMM'99, pp. 175-187, 1999.
- [7] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", in Proc. SIGCOMM'00, pp. 43-56, 2000.
- [8] E. Kohler, M. Handley and S. Floyd, IETF RFC 4340 "Datagram Congestion Control Protocol (DCCP)", March 2006.
- [9] C. Bouras and N. Kanakis, "Online AL-FEC protection over mobile unicast services", European Conference on Networks and Communications(EuCNC), pp. 229-233, 2015
- [10] <https://www.i-juse.co.jp/statistics/jirei/sympo/10/arima-model.html> (参照 2018-07-10)
- [11] <https://iperf.fr/iperf-download.php> (参照 2018-07-08)
- [12] HTB Linux queuing discipline manual - user guide, <http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm> (参照 2018-07-08)